



INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* BETIM
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Marcos Paulo Santos Nascimento

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA AUXÍLIO NO
ACOMPANHAMENTO SEMANAL DE TAREFAS DE LIMPEZA BASEADO NA
METODOLOGIA 5S EM UMA INDÚSTRIA DE AUTOPEÇAS: UM ESTUDO DE
CASO**

Betim

2024

Marcos Paulo Santos Nascimento

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA AUXÍLIO NO
ACOMPANHAMENTO SEMANAL DE TAREFAS DE LIMPEZA BASEADO NA
METODOLOGIA 5S EM UMA INDÚSTRIA DE AUTOPEÇAS: UM ESTUDO DE
CASO**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Arthur
Hermano Rezende Rosa

Betim

2024

FICHA CATALOGRÁFICA

N244d Nascimento, Marcos Paulo Santos
Desenvolvimento de uma aplicação Web para auxílio no acompanhamento semanal de tarefas de limpeza baseado na metodologia 5S em uma indústria de autopeças: um estudo de caso / Marcos Paulo Santos Nascimento. – 2024.

63 f. : il.

Trabalho de conclusão de curso (Bacharelado em Engenharia de Controle e Automação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Câmpus Betim, 2024.

Orientador: Prof. Dr. Arthur Hermano Rezende Rosa

1. Aplicação Web. 2. Metodologia 5S. 3. Gestão de cronograma. 4. Engenharia de Controle e Automação. I. Nascimento, Marcos Paulo Santos. II. Título.

CDU: 681.5



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Betim
Diretoria de Ensino
Docentes Automação Industrial e Tecnologia da Informação
Rua Itamarati - CEP 32677-564 - Betim - MG
3135976360 - www.ifmg.edu.br

Ata da defesa de TCC - 15 de março de 2024

No dia 15 de março de 2024, às 17h40 horas, o aluno Marcos Paulo Santos Nascimento, do curso de Engenharia de Controle e Automação do IFMG – Campus Betim, defendeu o Trabalho de Conclusão de Curso (TCC) intitulado:

"Desenvolvimento de uma Aplicação Web para Auxílio no Acompanhamento Semanal de Tarefas de Limpeza baseado na Metodologia 5S em uma Indústria de Autopeças: Um Estudo de Caso"

e foi **aprovado** com a **nota 66**, que está condicionada ao cumprimento dos procedimentos pós-defesa do TCC. Caso seja aprovado, deverá apresentar o trabalho com as devidas correções até o final do semestre letivo corrente e, feitas as correções, tem mais 10 dias corridos para entregar as cópias em meio eletrônico.

O aluno está ciente de que, caso não cumpra os procedimentos pós-defesa de TCC até a data estipulada, sua nota será considerada zero e a sua defesa de TCC será considerada nula. Também está ciente de que o trabalho de conclusão de seu curso poderá ser divulgado pela Instituição através dos seus meios de comunicação.

Nada mais havendo a tratar a reunião foi encerrada às 18h40 horas.

Betim, 15 de março de 2024.



Documento assinado eletronicamente por **Arthur Hermoano Rezende Rosa, Professor**, em 15/03/2024, às 20:32, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Cristiano Martins Nunes, Professor Substituto**, em 18/03/2024, às 10:11, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Bruno Porto Caetano, Professor Substituto**, em 26/03/2024, às 12:11, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **1874245** e o código CRC **D214AC61**.

23792.000490/2024-56

1874245v1

Dedico esta monografia a minha querida família, que sempre esteve ao meu lado, apoiando-me em todas as etapas desta jornada acadêmica. O amor, o incentivo e a paciência que vocês demonstraram são inestimáveis e fundamentais para a conclusão deste trabalho. Cada um de vocês desempenhou um papel essencial na minha vida, e esta monografia é um tributo ao apoio e amor que recebi ao longo de todos esses anos.

À minha amada mãe, Claudia Roberta dos Santos Nascimento, agradeço por ser a minha fonte constante de inspiração e por me ensinar o verdadeiro significado da dedicação e da perseverança. Sua sabedoria e amor são os pilares da minha vida acadêmica.

Ao meu querido pai, Carlos Heleno Nascimento, agradeço por sempre me encorajar a perseguir meus sonhos e por me mostrar o poder da determinação. Seu apoio inabalável é a força motriz por trás do meu sucesso.

Ao meu amado irmão, Michael Felipe Santos Nascimento, agradeço por ser meu amigo, meu companheiro e por compartilhar comigo as alegrias e desafios dessa jornada. Sua presença é uma bênção que enriquece minha vida.

À minha adorável avó, Maria Luzia, e ao meu amado avô, Adelson, agradeço por serem os guardiões da tradição e da sabedoria em nossa família. Suas histórias e conselhos moldaram a pessoa que me tornei e me inspiraram a buscar o conhecimento.

AGRADECIMENTOS

Gostaria de estender meus sinceros agradecimentos a todos os meus professores, amigos e colegas que me apoiaram ao longo desta jornada acadêmica. Suas orientações, insights e encorajamento foram cruciais para o sucesso deste trabalho.

Agradeço também Brebbo que forneceu todo o suporte necessário para realização deste estudo. Sem esses recursos, esta monografia não teria sido possível.

Por fim, agradeço a mim mesmo, por nunca desistir e por sempre acreditar que eu poderia alcançar meus objetivos acadêmicos. Esta monografia é a culminação de anos de trabalho árduo e dedicação, e estou imensamente grato por todas as experiências que me trouxeram até este momento.

Muito obrigado a todos que contribuíram de alguma forma para o desenvolvimento e conclusão deste trabalho. Esta monografia é dedicada a vocês, com profundo amor e gratidão.

"O conhecimento é a única riqueza
que cresce quando é compartilhada." -
Khalil Gibran

RESUMO

Este trabalho propõe o desenvolvimento de uma aplicação web para o gerenciamento de cronogramas de limpeza em ambientes corporativos, visando aprimorar a eficiência comunicativa entre equipes de limpeza, líderes de turno e analistas de produção. Através da utilização de métodos de desenvolvimento web, foram criados formulários para inserção e edição de tarefas, bem como recursos para visualização e exportação de dados em formato Excel. Os resultados demonstraram uma melhoria significativa na gestão dos cronogramas de limpeza, com a aplicação permitindo uma comunicação mais ágil e eficaz entre as equipes e uma adaptação mais eficiente às demandas em constante mudança. As auditorias de 5S indicaram um aumento nas notas de avaliação, evidenciando a eficácia da abordagem adotada. Conclui-se que a implementação do sistema proporcionou maior flexibilidade na gestão das tarefas de limpeza, melhorando a produtividade e a satisfação dos usuários.

Palavras-chave: Gerenciamento de cronograma; limpeza; aplicação web; 5S;

ABSTRACT

This work proposes the development of a web application for the management of cleaning schedules in corporate environments, aiming to improve communicative efficiency among cleaning teams, shift leaders, and production analysts. Through the use of web development methods, forms were created for the insertion and editing of tasks, as well as resources for visualization and data export in Excel format. The results demonstrated a significant improvement in the management of cleaning schedules, with the application allowing for more agile and effective communication between teams and a more efficient adaptation to constantly changing demands. The 5S audits indicated an increase in evaluation scores, highlighting the effectiveness of the adopted approach. It is concluded that the implementation of the system provided greater flexibility in the management of cleaning tasks, improving productivity and user satisfaction.

Keywords: Schedule Management; Cleaning; Web Application; 5S.

LISTA DE ILUSTRAÇÕES

Figura 1- Critérios de avaliação da auditoria de 5S.....	21
Figura 2 - Apresentação de instrução de trabalho para limpeza de piso e bancada .	23
Figura 3 – Apresentação do Firebase	25
Figura 4 – Google Agenda	28
Figura 5 – Controle de usuários do cronograma de limpeza.....	30
Figura 6 – Repositório do cronograma no Github.....	32
Figura 7 Recurso de Inspeccionar Elemento do Google Chrome	33
Figura 8 – <i>Login</i> de usuários e administradores cadastrados	34
Figura 9 Tela de registro de novos usuários	35
Figura 10 e-mail de recuperação.....	36
Figura 11 notificação de e-mail para recuperação	36
Figura 12 – Validação de formato de e-mail.....	37
Figura 13 - cronograma de limpeza.....	38
Figura 14 Botão exportar para o Excel.....	39
Figura 15 página de inserção de atividades.....	40
Figura 16 edição e remoção de atividade	41
Figura 17 formulário HTML de login	41
Figura 18 importação da estilização e scripts.....	42
Figura 19 Script página de login.....	43
Figura 20 Continuação do script página de login	45
Figura 21 código css página de login	46
Figura 22 página de registro de usuário	48
Figura 23 funcionalidade de registro de usuários.....	50
Figura 24 funcionalidade de registro de usuários.....	51
Figura 25 estilização registro de usuário.....	53
Figura 26 Indicador das auditorias de 5S.....	56
Figura 27 – Planta Brembo betim.....	62

LISTA DE TABELAS

Tabela 1 comparativo de sistemas.....	58
---------------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

- IFMG** - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais
- ABNT** - Associação Brasileira de Normas Técnicas
- NBR** - Normas Técnicas Brasileiras
- TCC** - Trabalho de Conclusão de Curso
- NR** - Norma Regulamentadora
- MTE** - Ministério do Trabalho e Emprego
- FAO** – Folha de análise de operação
- PDCA** - Plan-Do-Check-Act (um método de gestão para melhoria contínua)
- CSS** - Cascading Style Sheets (Folhas de Estilo em Cascata)
- HTML** - Hypertext Markup Language (Linguagem de Marcação de Hipertexto)
- PHP** - Hypertext Preprocessor (Pré-processador de Hipertexto)
- API** - Application Programming Interface (Interface de Programação de Aplicações)
- UI** - User Interface (Interface do Usuário)

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa.....	15
1.2	Colocação do Problema.....	15
1.3	Objetivos	15
1.4	Organização do Trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Planejamento de limpeza industrial.....	17
2.2	Desenvolvimento de software.....	18
3	METODOLOGIA	19
3.1	Recursos	20
3.1.1	Auditoria de 5S	20
3.1.2	Instruções de Trabalho.....	22
3.1.3	Ferramentas para criação da aplicação web	24
3.1.4	Firebase	Erro! Indicador não definido.
3.1.4.1	<i>Armazenamento em Tempo Real.....</i>	<i>25</i>
3.1.4.2	<i>Autenticação de Usuários.....</i>	<i>26</i>
3.1.4.3	<i>Firebase Hosting:</i>	<i>27</i>
3.1.4.4	<i>Integração com JavaScript.....</i>	<i>27</i>
3.1.4.5	<i>Firestore para Consultas e Indexação.....</i>	<i>27</i>
3.1.4.6	<i>Implementação de Notificações em Tempo Real</i>	<i>27</i>
3.1.5	Google Chrome	28
3.1.6	Desenvolvimento de telas	28
3.1.7	Níveis de acesso do sistema web	29
4	RESULTADOS E DISCUSSÃO	32
4.1	Desenvolvimento do aplicativo Web	32
4.1.1	Apresentação do aplicativo Web	33
4.2	Código-fonte	41
4.2.1	Formulário de login.....	41
4.3	Avaliação 5S	55
5	CONCLUSÃO	57
5.1	Problemas Encontrados	58

5.2	Trabalhos futuros	59
5.3	Inclusão de Links na Aplicação	60
5.4	Inclusão de um Modelo de Planta Baixa	61
5.5	Criação de Indicadores Futuros.....	62
	REFERÊNCIAS.....	64

1 INTRODUÇÃO

O desenvolvimento web vem ganhando cada vez mais destaque em diversas áreas, e a gestão de processos é uma área que tem se beneficiado muito com essa tecnologia. Pensando nisso, este trabalho de conclusão de curso tem como objetivo apresentar uma programação semanal para acompanhamento da limpeza na indústria de usinagem utilizando tecnologias web.

1.1 Justificativa

A manutenção da limpeza externa em ambientes industriais, especialmente no que se refere a pisos e máquinas, é crucial para assegurar não apenas a estética e a organização do local, mas também para garantir a segurança no trabalho e a longevidade dos equipamentos. A acumulação de resíduos e a falta de limpeza regular podem levar a riscos de acidentes, como escorregões e quedas, além de contribuir para o desgaste precoce de máquinas. Assim, a implementação de um cronograma semanal de limpeza representa uma estratégia fundamental para promover um ambiente de trabalho mais seguro, eficiente e agradável, além de enfatizar a importância da preservação ambiental e do bem-estar dos colaboradores.

1.2 Colocação do Problema

Nos processos industriais, a gestão da limpeza externa de pisos e máquinas frequentemente enfrenta desafios relacionados à eficácia e à coordenação entre os envolvidos. A falta de um cronograma estruturado para limpeza pode resultar em inconsistências na manutenção, afetando diretamente a segurança do ambiente de trabalho e a eficiência operacional. Ademais, a inexistência de uma rotina de limpeza contribui negativamente para a imagem da empresa e para a saúde dos funcionários, reforçando a necessidade de um sistema organizado que assegure a regularidade e a qualidade da limpeza externa.

1.3 Objetivos

O objetivo principal deste trabalho é apresentar um cronograma semanal para acompanhamento de limpeza em uma indústria de usinagem utilizando tecnologias

web, visando uma gestão aprimorada de limpeza e torná-la mais eficiente e segura. Para isso, serão definidos os seguintes objetivos específicos:

- Identificar as principais etapas do processo de limpeza na indústria de usinagem em questão;
- Definir as melhores práticas para a gestão de limpeza nesse contexto;
- Desenvolver uma plataforma web para o acompanhamento e registro das atividades de limpeza;
- Testar e validar a eficiência do cronograma semanalmente de limpeza em um ambiente real de produção.

1.4 Organização do Trabalho

Este trabalho está garantido em quatro capítulos. No primeiro capítulo, é apresentada a introdução do trabalho, incluindo a justificativa, colocação do problema, objetivos e organização do trabalho. No segundo capítulo, é feita uma revisão da literatura sobre o tema, abordando conceitos relacionados à gestão de limpeza em usinagem e tecnologias web. No terceiro capítulo, é apresentada a metodologia utilizada para o desenvolvimento do cronograma semanal de limpeza e da plataforma web. Por fim, no quarto capítulo, são apresentados os resultados obtidos com a implementação do cronograma e da plataforma, bem como as tentativas e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica do trabalho, demonstrando os principais conceitos para a elaboração e desenvolvimento de um software para atender e entregar melhorias no planejamento e execução da atividade de limpeza em uma indústria de usinagem.

2.1 Planejamento de limpeza industrial

A limpeza industrial é uma atividade importante para garantir a segurança e a qualidade do ambiente de trabalho, além de contribuir para a manutenção e o bom funcionamento dos equipamentos e maquinários. Para que a limpeza seja eficaz, é necessário que ela seja realizada de forma regular e organizada, com base em práticas recomendadas e normas técnicas.

De acordo com a Norma Regulamentadora (NR) nº 1, do Ministério do Trabalho e Emprego (MTE), cabe ao empregador implementar medidas de prevenção de acidentes e doenças ocupacionais, garantindo a segurança e a saúde dos trabalhadores. A NR nº 1 também estabelece que as medidas devem ser baseadas em práticas reconhecidas pela tecnologia, pela ciência e pela experiência.

A limpeza industrial deve ser realizada levando em consideração os riscos presentes em cada ambiente de trabalho. Por exemplo, em ambientes com risco de explosão, é necessário utilizar equipamentos e produtos específicos que não apresentem risco de ignição. Além disso, é importante utilizar produtos que sejam eficazes para a remoção dos tipos de sujidades presentes.

Para a realização da limpeza, é fundamental seguir as normas técnicas e as práticas recomendadas. A Associação Brasileira de Normas Técnicas (ABNT) possui diversas normas que tratam da limpeza industrial, como a ABNT NBR 14644-1, que estabelece os requisitos para salas limpas e ambientes controlados. Além disso, existem diversas publicações científicas e livros que tratam do assunto, apresentando técnicas e procedimentos para a limpeza industrial.

2.2 Desenvolvimento de software

A inclusão dos sistemas informatizados nas indústrias, tornou-se algo primordial para eficácia da execução das atividades, pois permite a acessibilidade e organização de dados relativos a processos tanto interno e externo da empresa.

O mercado atual existe em um mundo globalizado e a internet se tornou um componente essencial desse novo sistema comercial. O número de usuários, serviços, informações e facilidades disponíveis online cresce a cada dia. O desenvolvimento de software para Web difere do desenvolvimento de software para desktop em termos de público-alvo, acesso e manutenção. A forma de acesso torna-se mais acessível por ser um sistema web que pode ser acessado via navegadores de internet e possui características multiplataforma, pois não exige que o usuário final possua um sistema operacional específico, mas apenas um navegador de internet. Também é mais fácil de manter e atualizar porque não exige que o usuário final execute procedimentos de instalação, simplificando o suporte.

PHP, JavaScript, HTML e CSS são tecnologias fundamentais para a construção de websites. Cada uma delas desempenha um papel específico no processo de desenvolvimento web.

O HTML (Hypertext Markup Language) é a linguagem utilizada para criar a estrutura e o conteúdo de um website. Ele define a hierarquia de títulos, parágrafos, listas, imagens e outros elementos que compõem a página. O HTML é responsável pela marcação semântica do conteúdo, ou seja, ele fornece informações sobre a estrutura do texto e seu significado, permitindo que os mecanismos de busca e outros sistemas possam interpretá-lo corretamente.

O CSS (Cascading Style Sheets) é a linguagem utilizada para definir a aparência visual de um website. Ele permite definir as cores, fontes, tamanhos e posicionamento dos elementos na página. O CSS é responsável pela separação entre a apresentação visual e o conteúdo estrutural, permitindo que os designers possam criar layouts e estilos personalizados sem afetar a estrutura do HTML.

O JavaScript é uma linguagem de programação que permite adicionar interatividade e dinamismo aos websites. Ele é utilizado para criar efeitos de animação, validar formulários, criar pop-ups e outras funcionalidades que melhoram a

experiência do usuário. O JavaScript também permite integrar o website a serviços externos, como APIs de redes sociais ou sistemas de gerenciamento de conteúdo.

O PHP (Hypertext Preprocessor) é uma linguagem de programação *server-side* utilizada para criar websites dinâmicos e interativos. Ele é executado no servidor web e é responsável por processar formulários, enviar e-mails, gerar páginas personalizadas com base nas informações do usuário e outras funcionalidades complexas. O PHP é comumente utilizado em conjunto com bancos de dados, permitindo a criação de sistemas de gerenciamento de conteúdo e e-commerce.

Juntos, o HTML, CSS, JavaScript e PHP formam a base da maioria dos websites modernos. Eles são essenciais para a criação de websites visualmente atraentes, interativos e funcionais, incluindo o cronograma semanal web para acompanhamento de limpeza industrial. É importante ter um bom conhecimento dessas tecnologias para criar um website eficiente e bem estruturado.

3 METODOLOGIA

Neste estudo, adota-se uma metodologia integrada que enfatiza a eficácia da limpeza e do cronograma por meio da auditoria de 5S, reconhecida por sua capacidade de promover organização, limpeza e eficiência no ambiente de trabalho. A aplicação desses princípios visa otimizar os processos operacionais da aplicação web.

Selecionam-se ferramentas tecnológicas avançadas, como o Firebase, para o backend, devido à sua escalabilidade e facilidade de integração. O Google Chrome é utilizado como ambiente principal de testes, assegurando compatibilidade e desempenho. Prioriza-se o desenvolvimento de telas intuitivas e responsivas, buscando uma experiência de usuário superior.

Implementa-se, também, um sistema de níveis de acesso diferenciados na aplicação web, garantindo que cada usuário acesse apenas as funcionalidades relevantes ao seu papel, vital para a segurança dos dados e eficiência operacional.

3.1 Recursos

3.1.1 Auditoria de 5S

A abordagem metodológica de auditoria 5S é uma ferramenta utilizada para avaliar a eficiência da limpeza e outros requisitos dos cinco sentidos. O método 5S é originário do Japão e baseia-se em cinco princípios básicos que começam com a letra "S": *Seiri* (Senso de Utilização), *Seiton* (Senso de Organização), *Seiso* (Senso de Limpeza), *Seiketsu* (Senso de Padronização) e *Shitsuke* (Senso de Disciplina).

A auditoria de 5S envolve as seguintes etapas:

Planejamento: Definir os objetivos da auditoria e as áreas da indústria de usinagem que serão avaliadas. Estabeleça critérios de avaliação claros, como procedimentos de limpeza, organização de ferramentas e equipamentos, limpeza de áreas de trabalho, entre outros.

Preparação: Onde uma equipe é responsável pela auditoria e realiza um treinamento sobre os princípios e critérios do 5S. Elabore uma lista de verificação (*checklist*) que detalhe os itens a serem observados durante uma auditoria, como demonstrado na Figura 1- Critérios de avaliação da auditoria de 5S. Figura 1.

Figura 1- Critérios de avaliação da auditoria de 5S



Fonte: Gestão de 5S shopfloor management Brembo do Brasil, 2023.

Auditoria: Será realizado uma auditoria visitando as linhas de usinagem uma por uma, sendo utilizado uma lista de verificação para avaliar cada item conforme os

critérios considerados, registrando as observações e anotados os pontos fortes e fracos encontrados em relação à eficiência e a frequência de limpeza.

Análise dos resultados: Após a conclusão semanal de auditorias, será realizado uma reunião para definição de um plano de ação, buscando identificar áreas que apresentam boas práticas de limpeza e organização, bem como áreas que precisam ser melhoradas e problemas sistemático, priorizando os problemas identificados de acordo com seu custo e impacto na eficiência geral da limpeza.

Ações corretivas: Com base nos resultados da auditoria, será desenvolvido um plano de ação para melhorar a eficiência de limpeza nas áreas identificadas como deficientes. Estabelecendo metas e prazos para implementar as ações corretivas necessárias. Envolvendo uma equipe responsável e fornecendo treinamento adicional, se necessário.

Acompanhamento: Será necessário o monitoramento regular do progresso das ações corretivas integradas. Com reuniões de feedback das auditorias para avaliar se as melhorias estão sendo mantidas e se novas oportunidades de melhoria alcançadas.

Essa abordagem metodológica de auditoria 5S permite avaliar a eficiência de limpeza na fábrica de forma sistemática e identificar áreas que buscam melhorias. Além disso, promove a padronização e disciplina para manter um ambiente limpo e organizado, garantindo a eficiência geral das operações.




3.1.2 Instruções de Trabalho

A implementação das instruções de trabalho na indústria é um componente essencial para assegurar a eficácia das práticas de limpeza e manutenção autônoma. Estas instruções servem como um guia detalhado para os colaboradores, delineando as etapas específicas, os materiais necessários e a frequência com que as tarefas devem ser realizadas. Por exemplo, no caso da limpeza do piso, as instruções podem especificar a utilização de vassoura, pano, pá e água, bem como a necessidade de limpar a área ao final de cada turno de trabalho. Além disso, as instruções podem incluir medidas de segurança, como o uso de equipamentos de proteção individual e a correta manipulação de produtos de limpeza.

A clareza e a acessibilidade dessas instruções são fundamentais para garantir que todos os colaboradores compreendam e sigam corretamente os procedimentos

estabelecidos. Isso não apenas promove a manutenção da limpeza e da organização na indústria, mas também contribui para a prevenção de acidentes e a preservação da saúde dos trabalhadores. Portanto, as instruções de trabalho devem ser elaboradas de forma concisa e objetiva, sendo regularmente atualizadas para refletir as melhores práticas e as mudanças nos processos de limpeza.

Figura 2 - Apresentação de instrução de trabalho para limpeza de piso e bancada

		CHECK-LIST DE LIMPEZA E INSPEÇÃO MANUTENÇÃO AUTÔNOMA					LINGUAGEM PORTUGUES CODIGO: REV. Anexo 1 PROCO BB-02 02 NÚMERO 20					
PISO E BANCADA												
												
ATIVIDADE	STATUS DA MÁQUINA	DESCRIÇÃO	TURNO	S	T	Q	Q	S	S	D	TEMPO DESTINADO	OBSERVAÇÃO
LIMPEZA PISO	MAQ. TRABALHANDO	O QUE?: LIMPAR PISO DA LINHA EM QUE SE ESTÁ TRABALHANDO	1								6 MINUTOS	DENTRO DOS 10 MINUTOS DO FIM DO TURNO
		COMO?: COM VASSOURA, PANO, PÁ E ÁGUA.	2	CONFORME ROTINA DE LIMPEZA								
		PLANO DE REAÇÃO: AVISAR JOLLY OU LIDER CASO NÃO ENCONTRE MATERIAL PARA LIMPEZA.	3									
LIMPEZA BANCADA	MAQ. TRABALHANDO	O QUE?: LIMPAR BANCADA.	1								CICLO DE CONTROLE	
		COMO?: COM PANO, PÁ, ÁGUA.	2									
		PLANO REAÇÃO: AVISAR JOLLY OU LIDER CASO NÃO ENCONTRE MATERIAL PARA LIMPEZA.	3									

Fonte: Brembo do Brasil, 2023.

3.1.3 Ferramentas para criação da aplicação web

Para desenvolver o site de construção do cronograma web, foram utilizadas diversas bibliotecas e ferramentas para facilitar o processo de desenvolvimento e fornece recursos adicionais.

No *frontend*, foram empregadas as linguagens HTML, CSS e JavaScript para estruturar e estilizar as páginas do site. Para agilizar o desenvolvimento e garantir um layout responsivo.

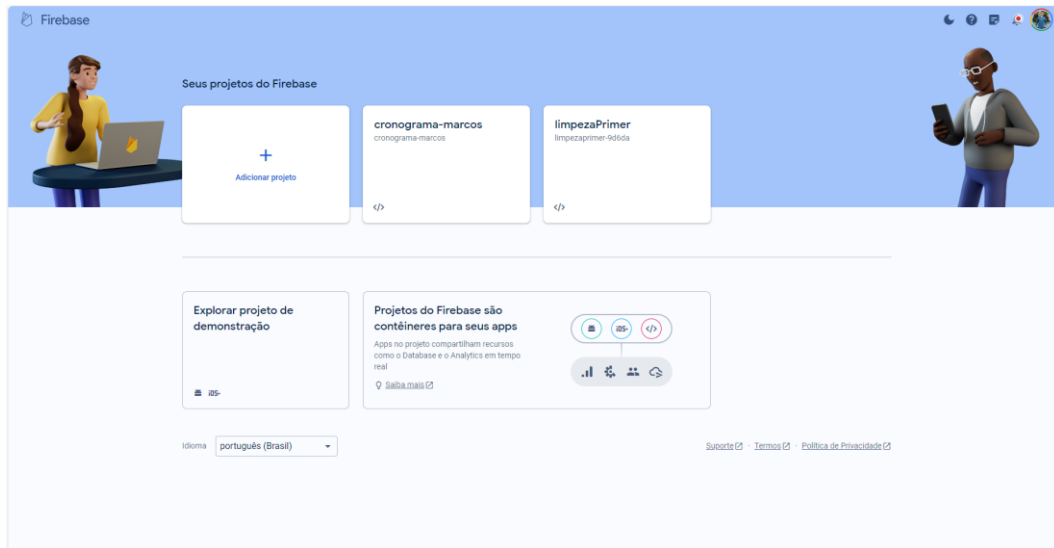
Essas tecnologias e bibliotecas foram selecionadas com base em suas capacidades e recursos, visando atender aos requisitos de desenvolvimento do site de construção do cronograma web. A escolha das bibliotecas e ferramentas levou em consideração a eficiência, a produtividade e a facilidade de integração entre as diferentes camadas do sistema.

A utilização dessas bibliotecas e ferramentas proporcionou um ambiente de desenvolvimento moderno e robusto, permitindo a criação de um site funcional, responsivo e interativo. Por meio da combinação dessas tecnologias, foi possível alcançar os objetivos propostos no projeto, oferecendo aos usuários uma experiência intuitiva e eficiente no uso do cronograma web.

3.1.4 Firebase

A utilização do Firebase, uma plataforma de desenvolvimento mobile e web da Google, foi uma escolha estratégica para potencializar a eficiência e escalabilidade do sistema proposto. O Firebase oferece um conjunto abrangente de serviços que incluem armazenamento em tempo real, autenticação de usuários, hospedagem de aplicativos, entre outros recursos integrados.

Figura 3 – Apresentação do Firebase



Fonte: tela de projetos do Firebase, 2023.

3.1.4.1 Armazenamento em Tempo Real

O Firebase *Realtime Database* foi adotado para armazenar e sincronizar dados em tempo real entre os diferentes componentes do sistema. Este serviço baseado em nuvem permite que as equipes de limpeza, líderes de turno e analistas de produção acessem e atualizem o cronograma de limpeza de forma instantânea e colaborativa.

O *Firebase Realtime Database* permite criar aplicativos avançados e colaborativos, permitindo acesso seguro ao banco de dados diretamente do código do cliente. Os dados são persistidos localmente e, mesmo off-line, os eventos em tempo real continuam a disparar, proporcionando ao usuário final uma experiência responsiva. Quando o dispositivo recupera a conexão, o *Realtime Database* sincroniza as alterações de dados locais com as atualizações remotas que ocorreram enquanto o cliente estava off-line, mesclando quaisquer conflitos automaticamente.

O *Realtime Database* fornece uma linguagem de regras flexível e baseada em expressões, chamada *Firebase Realtime Database Security Rules*, para definir como seus dados devem ser estruturados e quando os dados podem ser lidos ou gravados. Quando integrado ao *Firebase Authentication*, os desenvolvedores podem definir quem tem acesso a quais dados e como podem acessá-los.

O *Realtime Database* é um banco de dados *NoSQL* e, como tal, possui otimizações e recursos diferentes em comparação com um banco de dados relacional.

A API do *Realtime Database* foi projetada para permitir apenas operações que podem ser executadas rapidamente, oferecendo as seguintes principais capacidades:

- Tempo real: Em vez de solicitações HTTP típicas, o *Firestore Realtime Database* usa sincronização de dados: sempre que os dados são alterados, qualquer dispositivo conectado recebe essa atualização em milissegundos. Forneça experiências colaborativas e imersivas sem pensar no código de rede.
- Desligada: Os aplicativos do *Firestore* permanecem responsivos mesmo quando estão *off-line* porque o SDK do *Firestore Realtime Database* mantém seus dados no disco. Depois que a conectividade for restabelecida, o dispositivo cliente receberá todas as alterações perdidas, sincronizando-o com o estado atual do servidor.
- Acessível a partir de dispositivos clientes: O *Firestore Realtime Database* pode ser acessado diretamente de um dispositivo móvel ou navegador da web; não há necessidade de um servidor de aplicativos. A segurança e a validação de dados estão disponíveis por meio das regras de segurança do *Firestore Realtime Database*, regras baseadas em expressões que são executadas quando os dados são lidos ou gravados.

3.1.4.2 Autenticação de Usuários

Para garantir a segurança e a personalização da experiência do usuário, o *Firestore Authentication* foi implementado. Esse serviço permite a autenticação de usuários de forma segura, ajudando a controlar o acesso às funcionalidades do sistema.

O *Firestore Authentication* fornece serviços de *back-end*, SDKs fáceis de usar e bibliotecas de IU prontas para autenticar usuários no seu aplicativo. Ele suporta autenticação usando senhas, números de telefone, provedores de identidade federados populares como Google, Facebook e Twitter e muito mais.

O *Firestore Authentication* se integra perfeitamente a outros serviços do *Firestore* e aproveita padrões do setor, como OAuth 2.0 e OpenID Connect, para que possa ser facilmente integrado ao *back-end*.

3.1.4.3 Firebase Hosting:

A hospedagem do aplicativo web foi realizada no Firebase *Hosting*, fornecendo uma infraestrutura confiável e escalável. Isso garante que a aplicação seja acessível de forma rápida e eficiente para todos os usuários, independentemente da sua localização.

3.1.4.4 Integração com JavaScript

A integração do Firebase com JavaScript, por meio da Firebase SDK, permitiu a manipulação e recuperação de dados de maneira eficaz. Isso possibilitou a criação de uma interface amigável e responsiva para os usuários, facilitando a interação com o sistema.

3.1.4.5 Firestore para Consultas e Indexação

O Firestore, um banco de dados NoSQL do Firebase, foi utilizado para consultas mais complexas e eficientes. Isso contribuiu para a obtenção de informações específicas do cronograma de limpeza de maneira rápida, otimizando a eficiência operacional.

3.1.4.6 Implementação de Notificações em Tempo Real

A funcionalidade de notificações em tempo real foi implementada por meio do *Firebase Cloud Messaging* (FCM). Essa característica possibilita a comunicação imediata sobre atualizações no cronograma, garantindo que todas as equipes estejam cientes de mudanças importantes.

A escolha estratégica do Firebase como infraestrutura para o sistema proporcionou uma abordagem eficaz e simplificada para o desenvolvimento, operação e escalabilidade da aplicação web de gerenciamento de cronogramas de limpeza. Essa decisão baseou-se na confiabilidade e flexibilidade oferecidas pelos serviços integrados do Firebase, contribuindo para a eficiência e sucesso do projeto.

3.1.5 Google Chrome

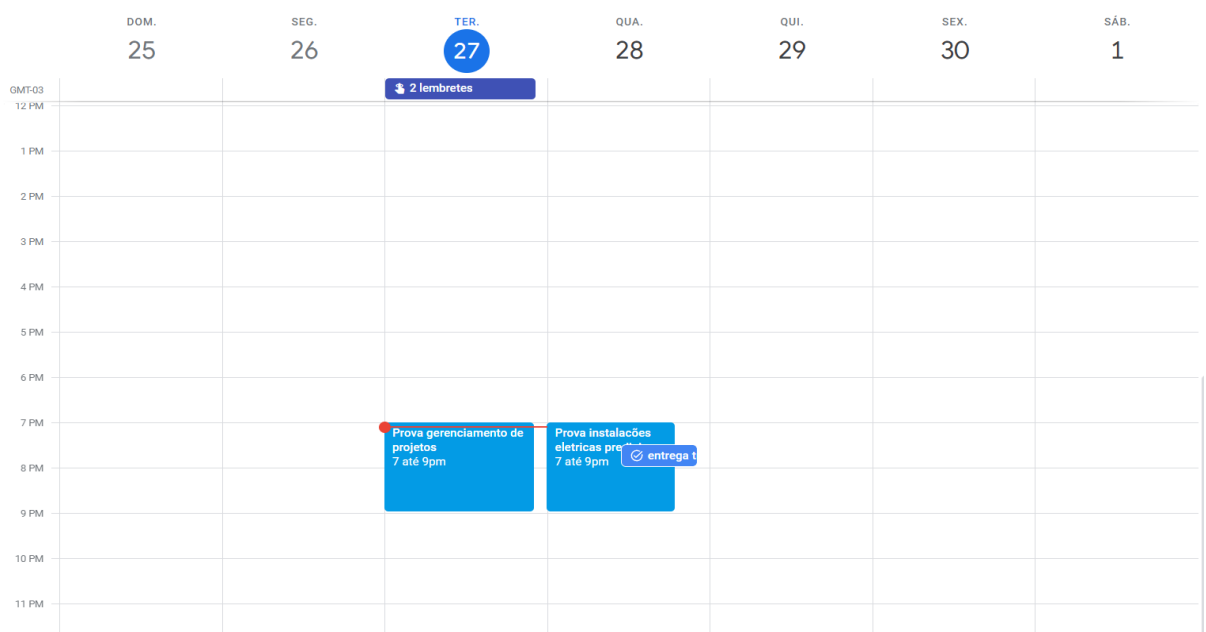
Google Chrome é um navegador para internet, criado pela Google, combina tecnologias sofisticadas com um design simples para tornar a Web mais rápida, mais segura e mais prática. (GOOGLE, 2014).

A inspeção de código é um recurso que permite visualizar e depurar códigos de uma página web. No Google Chrome, você pode acessar esse recurso clicando com o botão direito do mouse sobre o conteúdo web e selecionando a opção “Inspeccionar Elemento” ou pressionando a tecla de atalho “F12”. Com ele é possível escolher características visuais para o sistema, depurar códigos e acompanhar algumas funcionalidades.

3.1.6 Desenvolvimento de telas

O desenvolvimento da tela do projeto, que tem como tema o desenvolvimento de um cronograma web para acompanhamento de limpeza industrial, foi baseado em uma cuidadosa análise de diferentes softwares de agenda e cronograma, incluindo a renomada Google Agenda, Figura 4, bem como outras soluções existentes no mercado. Essas referências foram essenciais para compreender as melhores práticas e funcionalidades necessárias para atender aos requisitos da empresa em que trabalho.

Figura 4 – Google Agenda



Fonte: site Google Calendário, 2023.

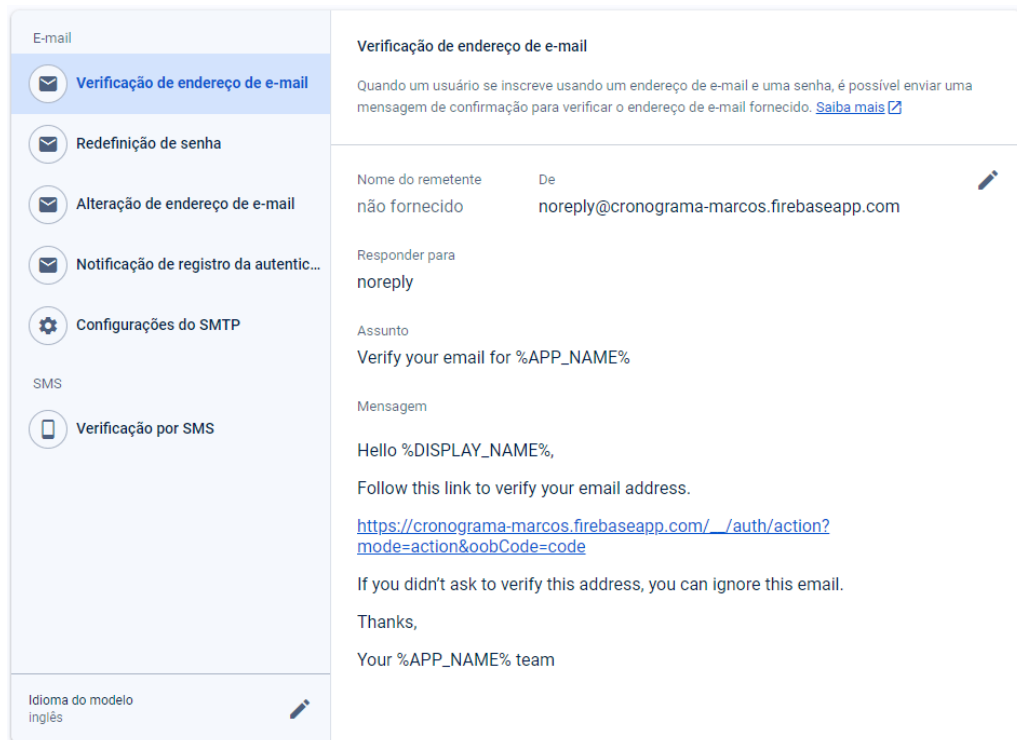
Ao estudar a Google Agenda e outros softwares similares, foi possível identificar os pontos fortes dessas ferramentas, como a capacidade de visualizar e gerenciar eventos e tarefas de forma intuitiva, a flexibilidade na criação de cronogramas personalizados. Esses recursos se mostraram extremamente relevantes para a proposta do meu projeto, pois a empresa necessita de uma solução eficiente para o acompanhamento de atividades de limpeza industrial em suas instalações.

3.1.7 Níveis de acesso do sistema web

A ideia de nível de acesso no projeto foi elaborada considerando a distinção de funções e responsabilidades entre o administrador, responsável pelo planejamento, e o usuário, que atua como executante das tarefas de limpeza.

No sistema desenvolvido, o administrador é designado como o usuário com maior nível de acesso. Essa função é atribuída a um indivíduo responsável pelo planejamento e coordenação geral das atividades de limpeza industrial. O administrador tem permissão para criar, editar e excluir tarefas, definir prazos e estabelecer prioridades. Além disso, ele pode visualizar o cronograma completo, acompanhar o progresso das tarefas e tomar decisões estratégicas com base nas informações fornecidas.

Figura 5 – Controle de usuários do cronograma de limpeza



Fonte: tela de autenticação do Firebase, 2023.

Por outro lado, o usuário é o responsável pela execução das tarefas de limpeza. Essa função é atribuída aos colaboradores encarregados de realizar as atividades práticas nos locais a serem limpos. Os usuários têm acesso restrito ao sistema, sendo capazes de visualizar as tarefas atribuídas a eles, bem como os prazos e as instruções pertinentes. Essa restrição de acesso garante que cada usuário tenha acesso apenas às informações relevantes para sua própria responsabilidade, evitando sobrecarga de informações e confusões.

A distinção de níveis de acesso é fundamental para garantir a eficiência e a segurança do sistema. Ao atribuir diferentes privilégios aos administradores e usuários, é possível manter um controle adequado sobre as atividades de limpeza industrial, assegurando que o planejamento seja realizado de forma estratégica e que a execução das tarefas ocorra conforme o planejado. Além disso, essa estrutura de acesso também promove a responsabilização dos envolvidos, permitindo que cada um cumpra seu papel dentro do processo de limpeza de maneira efetiva.

Dessa forma, a elaboração da ideia de nível de acesso no projeto busca garantir a divisão clara de responsabilidades e promover a eficiência e a segurança na gestão do cronograma de limpeza industrial. Com o administrador desempenhando um papel

de planejamento estratégico e o usuário concentrando-se na execução das tarefas, o sistema proporciona uma organização adequada e facilita a colaboração entre os envolvidos, contribuindo para o sucesso geral do processo de limpeza.

Dessa forma, as telas desenvolvidas no projeto foram concebidas com base nessas referências, visando fornecer uma interface amigável e de fácil navegação, além de incorporar funcionalidades relevantes para o contexto da limpeza industrial. Por exemplo, foi implementada a capacidade de criar e visualizar tarefas de limpeza, atribuir responsáveis, definir prazos e estabelecer prioridades. Também foi levada em consideração a necessidade de acompanhar o progresso das atividades por meio do campo de término da função, facilitando a análise e a tomada de decisões pelos gestores da empresa.

A inspiração obtida a partir desses softwares de agenda e cronograma deu um embasamento sólido para o desenvolvimento das telas do projeto, possibilitando a criação de uma interface intuitiva e eficiente, que atende às demandas específicas da empresa. O resultado é um sistema de acompanhamento de limpeza industrial que oferece uma visualização clara do cronograma, facilitando o planejamento, a execução e o monitoramento das atividades de limpeza, contribuindo para a otimização dos processos e o aumento da eficiência operacional da empresa.

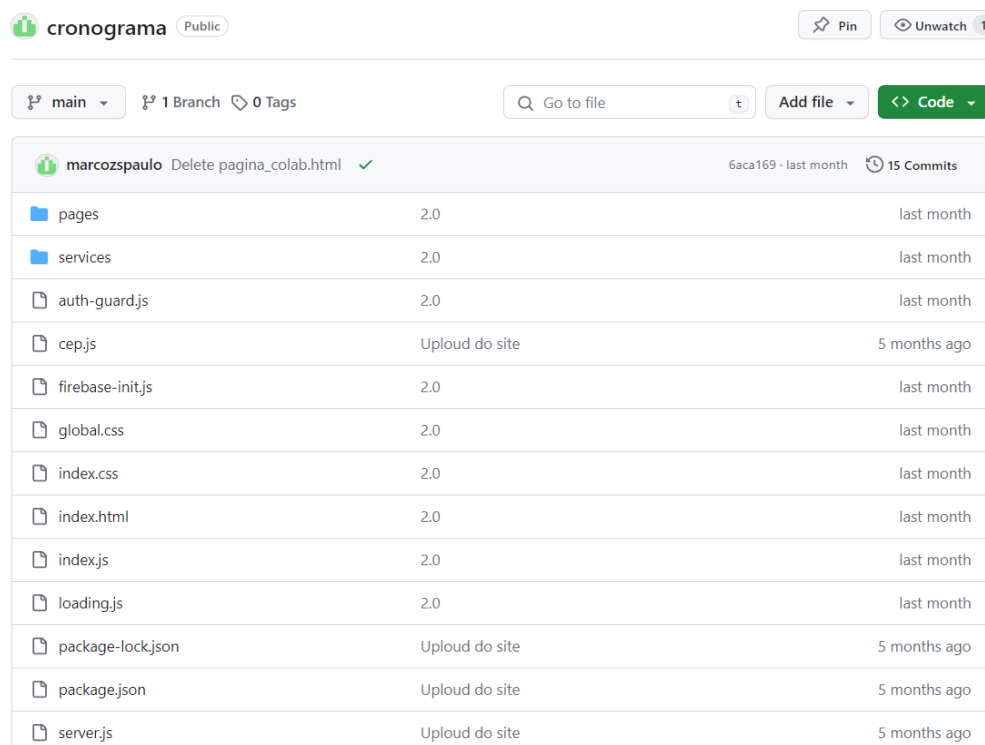
4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta uma análise abrangente da plataforma web desenvolvido, incluindo diagramas ilustrativos que destacam a utilização do sistema proposto e representações de classes e objetos relacionados ao projeto. Além disso, são detalhados os procedimentos de desenvolvimento do aplicativo, as interfaces de usuário, as funcionalidades-chave e alguns trechos de código que exemplificam as principais características do sistema.

4.1 Desenvolvimento do aplicativo Web

Para o desenvolvimento da aplicação web, o primeiro passo envolveu a instalação e configuração do ambiente de desenvolvimento. Inicialmente, foi realizado o download do navegador Google Chrome, que foi instalado e configurado com as opções padrão fornecidas pelo instalador. Em seguida, procedeu-se ao download e instalação do aplicativo Visual Studio Code e do GitHub, seguindo as instruções fornecidas pelos respectivos instaladores.

Figura 6 – Repositório do cronograma no Github



The screenshot shows the GitHub interface for a repository named 'cronograma'. At the top, it indicates the repository is 'Public' and has 15 commits. Below this, there are navigation options for 'main' branch, 1 branch, and 0 tags. A search bar and 'Add file' button are also visible. The main content is a list of files and folders with their respective commit dates.

File/Folder	Commit Date
pages	2.0 last month
services	2.0 last month
auth-guard.js	2.0 last month
cep.js	Uploud do site 5 months ago
firebase-init.js	2.0 last month
global.css	2.0 last month
index.css	2.0 last month
index.html	2.0 last month
index.js	2.0 last month
loading.js	2.0 last month
package-lock.json	Uploud do site 5 months ago
package.json	Uploud do site 5 months ago
server.js	Uploud do site 5 months ago

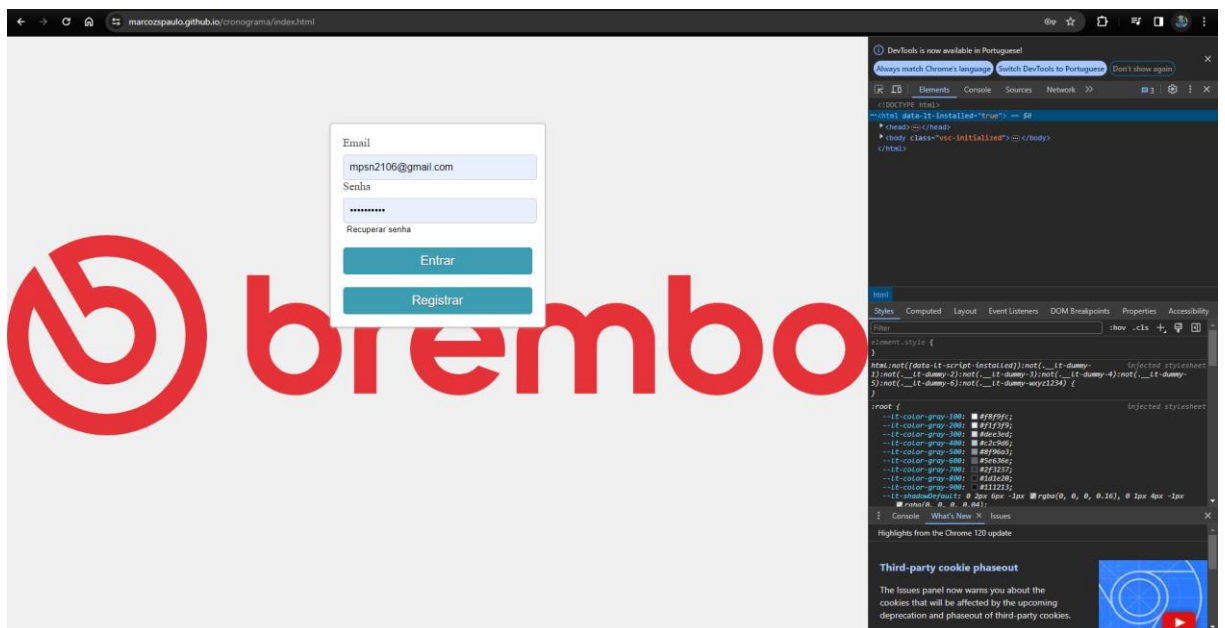
Fonte: GitHub, 2023.

Em seguida, procedeu-se à criação de todos os arquivos necessários para o funcionamento do sistema, com destaque para a apresentação de alguns códigos-chave e trechos significativos, que são exibidos na listagem A.

No decorrer do desenvolvimento do sistema, foram realizados vários testes para garantir o seu funcionamento. Para isso, foi utilizado o recurso de inspecionar elemento do Google Chrome. Esse recurso permite validar e acompanhar valores de variáveis JavaScript, testar métodos e fazer ajustes temporários no visual ou no comportamento do sistema.

A Figura 7 apresenta o recurso de Inspecionar elemento sendo utilizado:

Figura 7 Recurso de Inspecionar Elemento do Google Chrome



Fonte: Elaborado pelo próprio autor, 2023.

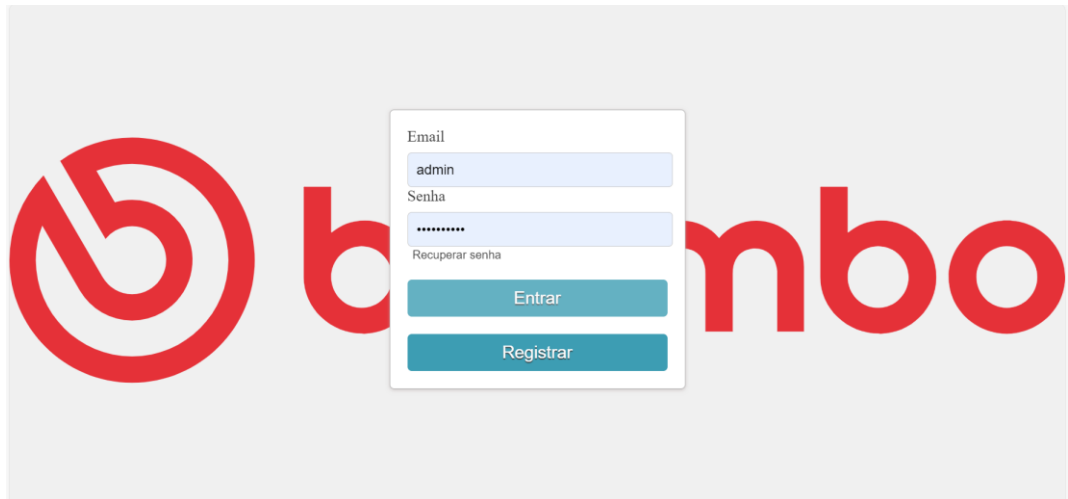
4.1.1 Apresentação do aplicativo Web

Neste tópico, serão apresentadas algumas telas do aplicativo web, produto do sistema objeto principal deste trabalho.

A primeira tela apresentada é a tela de *login*, que é uma interface de autenticação. Nessa tela, o usuário deve informar os valores de usuário e senha e pressionar o botão "Entrar" para possibilitar o acesso ao sistema. O acesso ao sistema

é permitido apenas a usuários cadastrados, que são divididos em dois níveis: administrador e colaborador.

Figura 8 – *Login* de usuários e administradores cadastrados



Fonte: Elaborado pelo próprio autor, 2023.

Ainda na tela inicial, oferecendo aos usuários a capacidade de efetuar o cadastro, proporcionando uma experiência intuitiva e direta. Ao clicar no botão "Registrar", os usuários são redirecionados para uma segunda tela, onde são apresentados três campos cruciais. O primeiro destina-se à inserção do e-mail desejado para o cadastro, seguido por campos para a definição da senha desejada e a confirmação dela. Dessa maneira, o processo de registro de um novo usuário é concluído de maneira simples e eficaz. Essa abordagem visa tornar o processo de cadastro acessível e amigável, garantindo uma experiência positiva para os usuários desde o início de sua interação com a aplicação.

Figura 9 Tela de registro de novos usuários

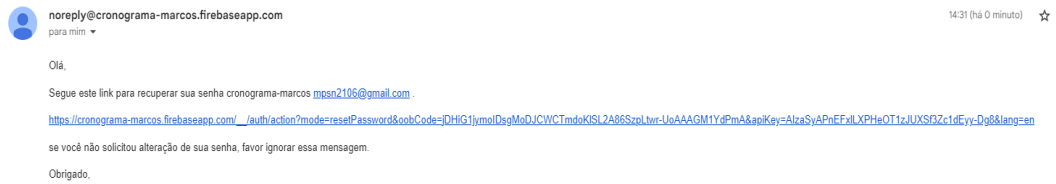


Fonte: Elaborado pelo próprio autor, 2023.

Para aprimorar a experiência do usuário e garantir a segurança no acesso à aplicação, implementamos um sistema avançado de recuperação de senha. Esta funcionalidade é essencial, considerando situações comuns em que os usuários podem esquecer suas credenciais. Ao selecionar a opção de recuperação de senha, o sistema automaticamente dispara um e-mail para o endereço registrado pelo usuário, ilustrado pela Figura 10. Este e-mail contém instruções claras e um link seguro para a redefinição da senha, garantindo assim que apenas o proprietário da conta tenha acesso à alteração da senha. A interface de usuário é projetada para fornecer feedback imediato, exibindo uma notificação na tela de login que confirma o envio do e-mail de recuperação, exibido na Figura 11, melhorando assim a transparência e a comunicação com o usuário.

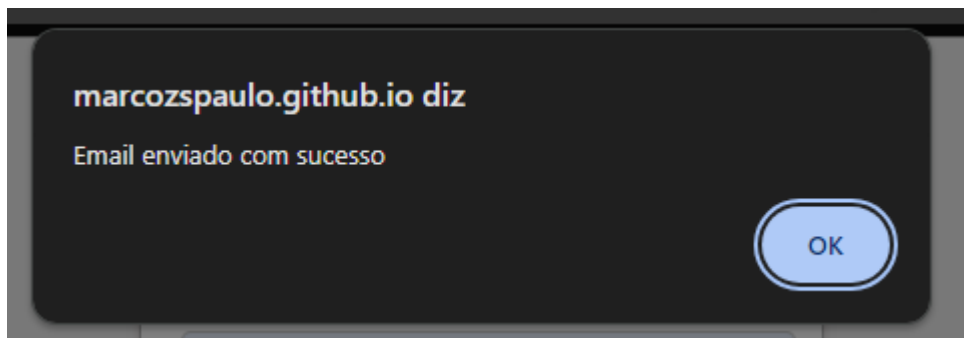
Essa abordagem visa fornecer aos usuários uma maneira segura e eficaz de recuperar o acesso à sua conta em caso de esquecimento da senha. A notificação na tela de login e o e-mail enviado proporcionam instruções claras e um processo simplificado para a troca de senha, contribuindo para uma experiência positiva do usuário em situações de recuperação de senha.

Figura 10 e-mail de recuperação



Fonte: Elaborado pelo próprio autor, 2023.

Figura 11 notificação de e-mail para recuperação



Fonte: Elaborado pelo próprio autor, 2023.

Além disso, enfatizamos a importância da validação de dados no processo de cadastro. A aplicação está configurada para identificar e rejeitar formatos inválidos de e-mail, orientando o usuário a fornecer um endereço válido. Esta etapa é crucial para manter a integridade dos dados e facilitar a comunicação efetiva. Por exemplo, ao tentar registrar-se com um formato de e-mail incorreto, a aplicação exibe uma mensagem de erro específica, como demonstrado na Figura 11. Esta funcionalidade não apenas previne erros comuns durante o cadastro, mas também educa os usuários sobre o formato adequado de e-mails, melhorando a qualidade dos dados inseridos no sistema.

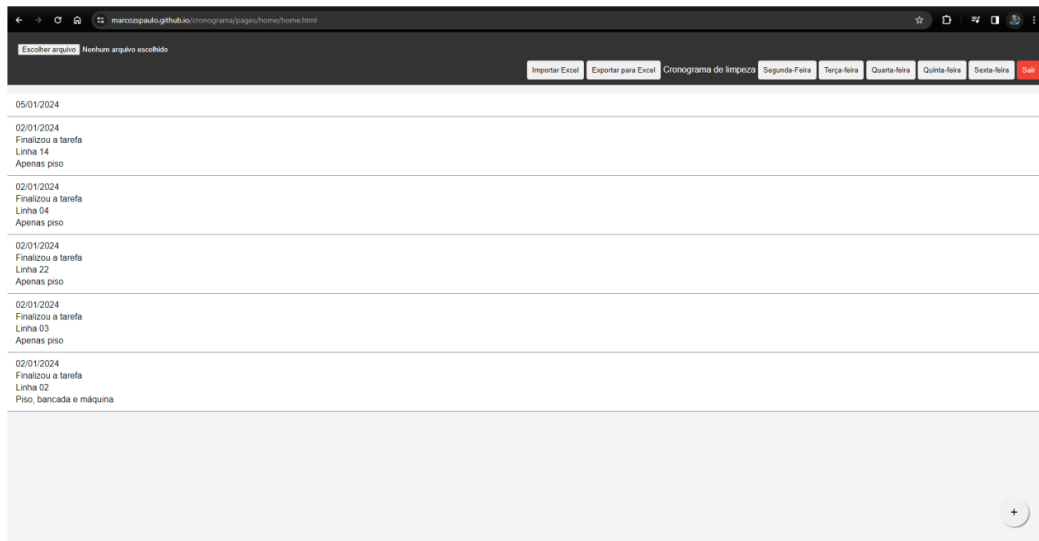
Figura 12 – Validação de formato de e-mail.

A imagem mostra uma interface de usuário para login. No topo, há o rótulo "Email" e um campo de entrada contendo o texto "admin". Abaixo do campo, uma barra vermelha exibe a mensagem de erro "Email é inválido". Logo abaixo, há o rótulo "Senha" e um campo de entrada com pontos para ocultar o texto. Abaixo do campo de senha, há o link "Recuperar senha". Na base da interface, há dois botões grandes e arredondados em um tom de azul-teal: "Entrar" e "Registrar".

Fonte: Elaborado pelo próprio autor, 2023.

Após a bem-sucedida autenticação do colaborador no sistema, a interface exibida na Figura 13, oferece uma experiência rica e de fácil acesso. Destacam-se elementos como os botões 'Sair', proporcionando uma saída segura do sistema. A principal funcionalidade desta tela é a apresentação do cronograma, organizando todas as atividades em ordem decrescente. Além disso, são disponibilizados botões específicos para cada dia útil da semana. Ao serem selecionados, esses botões revelam as atividades correspondentes ao dia da semana escolhido, dentro da semana atual. Por exemplo, ao selecionar a terça-feira na semana 01 de 2024, serão exibidas as atividades programadas para o dia 02 de janeiro de 2024.

Figura 13 - cronograma de limpeza

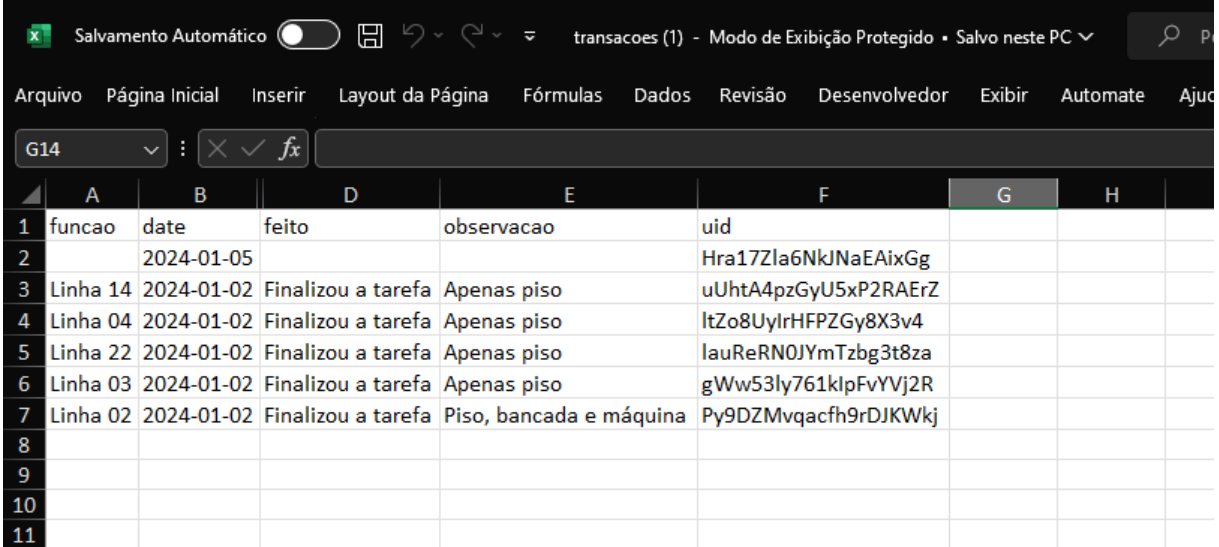


Fonte: Elaborado pelo próprio autor, 2023.

Outro destaque dessa interface é o botão de exportação para o Excel, que oferece aos usuários a capacidade de baixar uma tabela contendo informações detalhadas sobre as atividades, suas datas, status e observações associadas. O resultado dessa exportação pode ser visualizado na Figura 14, proporcionando uma visão clara e estruturada das informações do cronograma.

Essa abordagem visa otimizar a experiência do usuário, fornecendo uma navegação intuitiva e funcionalidades que atendem às necessidades práticas dos usuários, como a visualização detalhada das atividades, a exportação eficiente de dados e a saída segura do sistema.

Figura 14 Botão exportar para o Excel



	A	B	D	E	F	G	H
1	funcao	date	feito	observacao	uid		
2		2024-01-05			Hra17Zla6NkJNaEAixGg		
3	Linha 14	2024-01-02	Finalizou a tarefa	Apenas piso	uUhtA4pzGyU5xP2RAErZ		
4	Linha 04	2024-01-02	Finalizou a tarefa	Apenas piso	ltZo8UylrHFPZGy8X3v4		
5	Linha 22	2024-01-02	Finalizou a tarefa	Apenas piso	lauReRN0JYmTzbg3t8za		
6	Linha 03	2024-01-02	Finalizou a tarefa	Apenas piso	gWw53ly761klpFvYVj2R		
7	Linha 02	2024-01-02	Finalizou a tarefa	Piso, bancada e máquina	Py9DZMvqacfh9rDJKWkj		
8							
9							
10							
11							

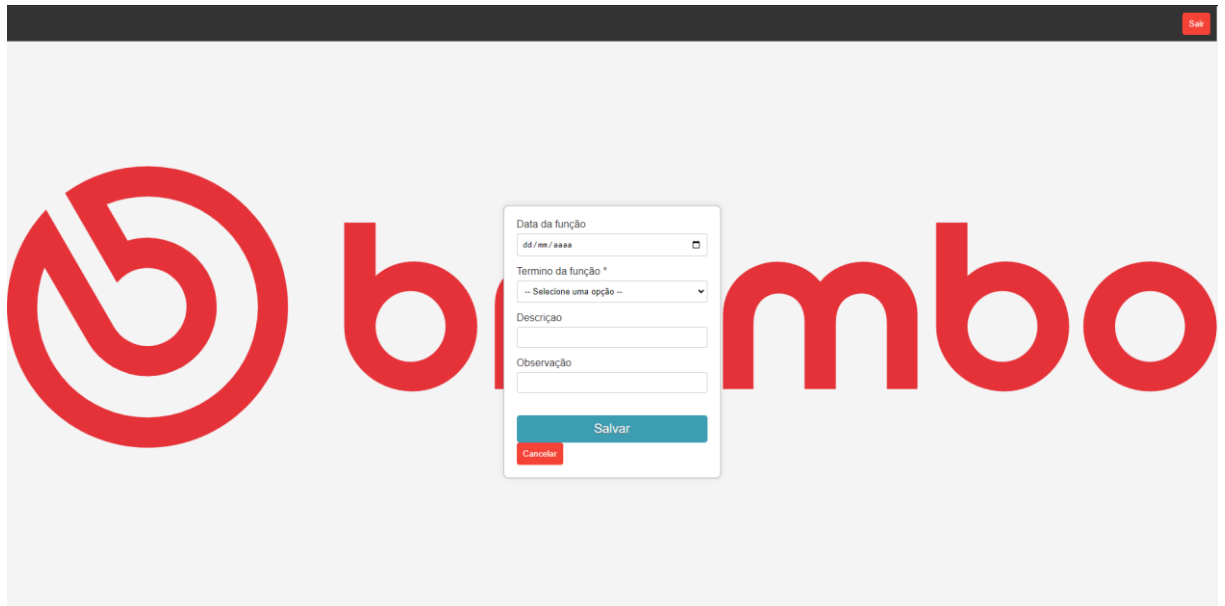
Fonte: Elaborado pelo próprio autor, 2023.

Na página inicial do cronograma de limpeza, destaca-se o botão "+" como uma funcionalidade adicional. Ao ser selecionado, direciona os usuários para a página de inserção de atividades, conforme demonstrado na Figura 15. Nessa página, os usuários têm a capacidade de detalhar as atividades de limpeza, fornecendo informações essenciais.

Na interface de inserção de atividades, os usuários podem especificar a data programada para a realização da atividade. Além disso, têm a opção de indicar se a atividade foi concluída ou não, fornecendo uma visão instantânea do status atual. A descrição da atividade é registrada, detalhando as tarefas planejadas. Adicionalmente, um campo de observações oferece a flexibilidade de incluir informações ou notas relevantes, ao final podendo salvar esse registro ou cancelar.

É fundamental destacar que todas as atividades inseridas por meio dessa funcionalidade são integradas ao cronograma principal, garantindo que todos os usuários tenham acesso às atualizações em tempo real. Essa abordagem visa simplificar e centralizar o processo de inserção de atividades, promovendo uma colaboração eficaz e mantendo todos os membros da equipe informados sobre as responsabilidades programadas. Essa integração contribui para a transparência e eficiência na gestão do cronograma de limpeza.

Figura 15 página de inserção de atividades

A imagem mostra a interface de usuário para a inserção de atividades no sistema bimbo. O formulário é centralizado e contém os seguintes elementos: um campo de data rotulado 'Data da função' com o formato 'dd/mm/aaaa' e um ícone de calendário; um menu suspenso rotulado 'Termino da função *' com a opção '-- Selecione uma opção --'; um campo de texto rotulado 'Descrição'; um campo de texto rotulado 'Observação'; um botão azul rotulado 'Salvar'; e um botão vermelho rotulado 'Cancelar'. O fundo da interface apresenta o logotipo 'bimbo' em vermelho.

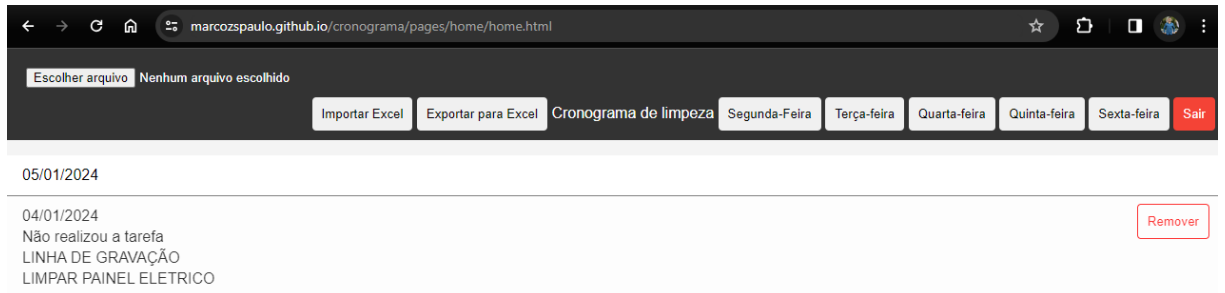
Fonte: Elaborado pelo próprio autor, 2023.

No menu inicial do cronograma, proporcionamos aos usuários a capacidade de editar atividades selecionadas, oferecendo uma abordagem intuitiva para gerenciamento eficiente. Ao passar o cursor sobre uma atividade, é possível selecioná-la para edição. Mais à direita, um botão de remoção se torna visível, permitindo a exclusão da atividade selecionada, conforme demonstrado na Figura 16.

Essa funcionalidade de edição oferece flexibilidade aos usuários, permitindo ajustes e atualizações conforme necessário. A opção de remoção proporciona uma maneira direta de eliminar atividades que não são mais relevantes ou necessárias no cronograma.

Essas características visam aprimorar a experiência do usuário, fornecendo meios eficazes para a gestão dinâmica e personalizada do cronograma de limpeza. A combinação de opções de edição e remoção facilita a manutenção e atualização contínuas do cronograma, contribuindo para a agilidade e organização na execução das tarefas planejadas.

Figura 16 edição e remoção de atividade



Fonte: Elaborado pelo próprio autor, 2023.

4.2 Código-fonte

4.2.1 Formulário de login

O código incorpora um formulário HTML destinado ao processo de *login*, apresentando campos para inserção de e-mail e senha. Esses campos são acompanhados por rótulos (labels) para fornecer uma orientação clara aos usuários.

Figura 17 formulário HTML de login

```

1  <!doctype html>
2
3  <html>
4
5  <head>
6    <title>Login</title>
7  </head>
8
9  <body >
10
11   <form>
12     <div>
13       <div><label class="character">Email</label></div>
14       <input type="email" name="email" id="email" placeholder="seu@email.com" onchange="onChangeEmail()"/>
15       <div class="error" id="email-required-error">Email é obrigatório</div>
16       <div class="error" id="email-invalid-error">Email é inválido</div>
17     </div>
18     <div><label class="character">Senha</label></div>
19     <input type="password" name="password" id="password" placeholder="Senha" onchange="onChangePassword()"/>
20     <div class="error" id="password-required-error">Senha é obrigatória</div>
21   </div>
22   <div>
23     <button type="button" class="clear" id="recover-password-button" disabled="true" onclick="recoverPassword()">Recuperar senha</button>
24   </div>
25   <div>
26     <button type="button" class="solid" id="login-button" disabled="true" onclick="login()">Entrar</button>
27   </div>
28   <div>
29     <button type="button" class="outline" onclick="register()">Registrar</button>
30   </div>
31 </form>
32 </body>
33 </html>

```

Fonte: Elaborado pelo próprio autor, 2023.

Para estilização e scripts foi feito:

- Importação de arquivos CSS (index.css e global.css) para aplicar estilos à página.

- Importação de scripts JavaScript (index.js, loading.js, e validations.js) para manipulação de comportamentos dinâmicos na interface.
- Importação das bibliotecas do Firebase e do script de inicialização do Firebase para suporte à autenticação.

Figura 18 importação da estilização e scripts

```
37 <link rel="stylesheet" href="index.css">
38 <link rel="stylesheet" href="global.css">
39
40 <script src="index.js"></script>
41 <script src="loading.js"></script>
42 <script src="validations.js"></script>
43
44 <script src="https://www.gstatic.com/firebasejs/10.4.0/firebase-app-compat.js"></script>
45 <script src="https://www.gstatic.com/firebasejs/10.4.0/firebase-auth-compat.js"></script>
46 <script src="firebase-init.js"></script>
47
```

Fonte: Elaborado pelo próprio autor, 2023.

O código JavaScript fornecido na página de login exerce várias ações em resposta às interações do usuário. Aqui estão as principais ações associadas ao código demonstrado na Figura 19:

Figura 19 Script página de login

```

1 //Email
2 function onChangeEmail() {
3     toggleButtonsDisable();
4     toggleEmailErrors();
5 }
6 //Senha
7 function onChangePassword() {
8     toggleButtonsDisable();
9     togglePasswordErrors();
10 }
11
12 //Autenticação de login e senha no banco de dados
13 function login() {
14     showLoading();
15     firebase.auth().signInWithEmailAndPassword(
16         form.email().value, form.password().value
17     ).then(() => {
18         hideLoading();
19         window.location.href = "pages/home/home.html";
20     }).catch(error => {
21         hideLoading();
22         alert(getErrorMessage(error));
23     });
24 }
25
26 //Direciona para tela de registrar
27 function register() {
28     window.location.href = "pages/register/register.html";
29 }
30
31 //Recuperar senha
32 function recoverPassword() {
33     showLoading();
34     firebase.auth().sendPasswordResetEmail(form.email().value).then(() => {
35         hideLoading();
36         alert('Email enviado com sucesso');
37     }).catch(error => {
38         hideLoading();
39         alert(getErrorMessage(error));
40     });
41 }
42
43 //Mensagem de erro de usuario ou senha invalida
44 function getErrorMessage(error) {
45     if (error.code == "auth/user-not-found") {
46         return "Usuário não encontrado";
47     }
48     if (error.code == "auth/invalid-login-credentials") {
49         return "Senha inválida";
50     }
51     return error.message;
52 }

```

Fonte: Elaborado pelo próprio autor, 2023.

- onChangeEmail () e onChangePassword():

Atualizam dinamicamente os erros associados ao e-mail e à senha.

Habilitam ou desabilitam os botões com base nas entradas do usuário.

- login():

Ao ser acionado pelo clique no botão "Entrar":

Mostra um indicador de carregamento.

Utiliza o Firebase Authentication para autenticar o usuário com o e-mail e senha fornecidos.

Se a autenticação for bem-sucedida:

Oculto o indicador de carregamento.

Redireciona o usuário para a página inicial ("pages/home/home.html").

Se houver um erro durante a autenticação:

Oculto o indicador de carregamento.

Exibe um alerta com a mensagem de erro específica.

- register():

Ao ser acionado pelo clique no botão "Registrar":

Redireciona o usuário para a página de registro ("pages/register/register.html").

- recoverPassword():

Ao ser acionado pelo clique no botão "Recuperar senha":

Mostra um indicador de carregamento.

Utiliza o Firebase Authentication para enviar um e-mail de redefinição de senha para o endereço fornecido.

Se o e-mail for enviado com sucesso:

Oculto o indicador de carregamento.

Exibe um alerta informando que o e-mail foi enviado com sucesso.

Se houver um erro no envio do e-mail:

Oculto o indicador de carregamento.

Exibe um alerta com a mensagem de erro específica.

- getErrorMessage(error):

Retorna mensagens de erro específicas com base nos códigos de erro do Firebase.

Figura 20 Continuação do script página de login

```

54 //Autenticação de email valido.
55 function toggleEmailErrors() {
56     const email = form.email().value;
57     form.emailRequiredError().style.display = email ? "none" : "block";
58
59     form.emailInvalidError().style.display = validateEmail(email) ? "none" : "block";
60 }
61
62
63 //Autenticação de senha valida
64 function togglePasswordErrors() {
65     const password = form.password().value;
66     form.passwordRequiredError().style.display = password ? "none" : "block";
67 }
68
69 //Botão de entrar desativado, ira ativar quando aplicar email e senha
70 function toggleButtonsDisable() {
71     const emailValid = isEmailValid();
72     form.recoverPasswordButton().disabled = !emailValid;
73
74     const passwordValid = isPasswordValid();
75     form.loginButton().disabled = !emailValid || !passwordValid;
76 }
77
78 //Email valido
79 function isEmailValid() {
80     const email = form.email().value;
81     if (!email) {
82         return false;
83     }
84     return validateEmail(email);
85 }
86
87 //Senha valida
88 function isPasswordValid() {
89     return form.password().value ? true : false;
90 }
91
92 const form = {
93     email: () => document.getElementById("email"),
94     emailInvalidError: () => document.getElementById("email-invalid-error"),
95     emailRequiredError: () => document.getElementById("email-required-error"),
96     loginButton: () => document.getElementById("login-button"),
97     password: () => document.getElementById("password"),
98     passwordRequiredError: () => document.getElementById("password-required-error"),
99     recoverPasswordButton: () => document.getElementById("recover-password-button"),
100 }

```

Fonte: Elaborado pelo próprio autor, 2023.

- toggleEmailErrors() e togglePasswordErrors():

Controlam dinamicamente a exibição de mensagens de erro relacionadas ao e-mail e à senha.

- toggleButtonsDisable():

Habilita ou desabilita dinamicamente os botões de login e recuperação de senha com base na validade do e-mail e da senha.

- isEmailValid() e isPasswordValid():

Verificam se o e-mail e a senha atendem a critérios específicos.

- form:

Fornece funções para acessar elementos do formulário por meio de seus IDs, simplificando a manipulação de elementos HTML no código.

O conjunto de regras CSS apresentado desempenha um papel fundamental na estilização dos elementos HTML relacionados aos campos de entrada de e-mail e senha em uma página web. A classe `.character` define estilos específicos para o texto associado, proporcionando um design claro e legível. Com uma combinação de tamanho de fonte apropriado, cor de texto adequada e espaçamento bem definido, a classe destaca-se ao orientar o usuário de maneira eficaz. Além disso, as regras aplicadas aos campos de entrada do tipo "email" e "password" garantem uma apresentação consistente e agradável. A largura generosa, o preenchimento interno, as bordas suavemente arredondadas e uma paleta de cores neutras contribuem para uma experiência visual coesa e atraente. Essas escolhas de estilo não apenas favorecem a estética da página, mas também otimizam a usabilidade, criando uma interface intuitiva e amigável para os usuários interagirem com eficiência.

Figura 21 código css página de login

```
1      .character {
2          display: block;
3          font-size: 18px;
4          color: #333;
5          margin-bottom: 8px;
6      }
7
8      input[type="email"],
9      input[type="password"]{
10         width: 95%;
11         padding: 10px;
12         font-size: 16px;
13         border: 1px solid #ccc;
14         border-radius: 5px;
15     }
16
```

Fonte: Elaborado pelo próprio autor, 2023.

O trecho de código CSS fornecido está estilizando elementos da página HTML relacionados a campos de entrada de e-mail e senha, possibilitando a seguinte análise:

1. **`.caracter``:**
 - **Descrição:** Define estilos para a classe `.caracter``.
 - **Propriedades:**
 - `display: block;`: Define que o elemento será exibido como bloco, ocupando toda a largura disponível.
 - `font-size: 18px;`: Define o tamanho da fonte como 18 pixels.
 - `color: #333;`: Define a cor do texto como um tom de cinza escuro (#333).
 - `margin-bottom: 8px;`: Adiciona uma margem inferior de 8 pixels para separação visual de elementos.

2. **`input[type="email"], input[type="password"]``:**
 - **Descrição:** Estiliza campos de entrada do tipo "email" e "password".
 - **Propriedades:**
 - `width: 95%;`: Define a largura do campo de entrada como 95% da largura disponível.
 - `padding: 10px;`: Adiciona um preenchimento interno de 10 pixels para melhorar o espaçamento interno.
 - `font-size: 16px;`: Define o tamanho da fonte como 16 pixels.
 - `border: 1px solid #ccc;`: Adiciona uma borda de 1 pixel sólido com uma cor cinza claro (#ccc).
 - `border-radius: 5px;`: Adiciona um pequeno raio de borda de 5 pixels para suavizar as bordas dos campos.

Essas definições visuais contribuem para uma apresentação mais limpa e legível dos campos de entrada na página HTML, seguindo as práticas comuns de design. A utilização de cores neutras e tamanhos de fonte legíveis promove uma experiência de usuário agradável.

Figura 22 página de registro de usuário

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <title> Registrar Usuário</title>
6  </head>
7
8  <body class="panel">
9    <form>
10     <div class="form-field">
11       <div><label>Email</label></div>
12       <input type="email" placeholder="seu@email.com"
13         onchange="onChangeEmail()" id="email"/>
14       <div class="error" id="email-required-error">Email é obrigatório</div>
15       <div class="error" id="email-invalid-error">Email é inválido</div>
16     </div>
17     <div class="form-field">
18       <div><label>Senha</label></div>
19       <input type="password" placeholder="Insira a sua senha" id="password" onchange="onChangePassword()"/>
20       <div class="error" id="password-required-error">Senha é obrigatória</div>
21       <div>
22         <div class="error" id="password-min-length-error">Senha deve ter pelo menos 6 caracteres</div>
23       </div>
24     </div>
25     <div class="form-field">
26       <div><label>Confirmar Senha</label></div>
27       <input type="password" placeholder="Confirmar a senha" id="confirmPassword" onchange="onChangeConfirmPassword()"/>
28       <div class="error" id="password-doesnt-match-error">Senha e Confirmar Senha deve ser iguais</div>
29     </div>
30     <div class="form-field">
31       <div>
32         <button type="button" class='solid' disabled="true" id="register-button" onclick="register()">Registrar</button>
33       </div>
34     </div>
35     <div>
36       <button type="button" class='outline' onclick="login()">Login</button>
37     </div>
38   </div>
39 </form>
40 </body>
41
42 <link rel="stylesheet" href="../../index.css">
43 <link rel="stylesheet" href="../../global.css">
44
45 <script src="registrar.js"></script>
46 <script src="../../validations.js"></script>
47 <script src="../../loading.js"></script>
48
49 <script src="https://www.gstatic.com/firebasejs/10.4.0/firebase-app-compat.js"></script>
50 <script src="https://www.gstatic.com/firebasejs/10.4.0/firebase-auth-compat.js"></script>
51 <script src="../../firebase-init.js"></script>
52
53 </html>

```

Fonte: Elaborado pelo próprio autor, 2023.

Este trecho de código HTML, Figura 22, é parte da interface de usuário de uma aplicação web para gerenciamento de cronograma de limpeza. Ele é responsável por criar a página de registro de usuário. Vamos analisar as principais partes do código:

1. **Estrutura Básica**: O código começa com a declaração do tipo de documento (`<!DOCTYPE html>`) seguida pela estrutura básica de uma página HTML, incluindo as tags `<html>`, `<head>` e `<body>`. Dentro do `<head>`, o título da página é definido como "Registrar Usuário".

2. **Formulário de Registro**: Dentro do `<body>`, há um formulário (`<form>`) que contém vários campos de entrada (`<input>`) para coletar informações do usuário, como email e senha. Cada campo de entrada está contido em uma `div` com a classe `form-field`, que provavelmente é usada para estilização e layout.

3. **Validação de Entrada**: Cada campo de entrada possui um atributo `onchange` que chama uma função JavaScript quando o valor do campo muda. Por exemplo, `onchange="onChangeEmail()"` para o campo de email. Isso sugere que a validação da entrada do usuário é feita dinamicamente. Além disso, mensagens de erro como "Email é obrigatório" e "Senha é obrigatória" estão presentes, mas inicialmente ocultas (provavelmente são exibidas quando um erro de validação ocorre).

4. **Botões**: Há dois botões no formulário: um para registrar e outro para fazer login. O botão de registrar (`<button type="button" id="register-button" onclick="register()>`) chama a função `register()` quando clicado, enquanto o botão de login chama a função `login()`.

5. **Links de Estilo e Scripts**: O documento HTML vincula a folhas de estilo externas (`index.css` e `global.css`) para definir a aparência da página. Também inclui vários scripts JavaScript (`registrar.js`, `validations.js`, `loading.js`, e os scripts do Firebase) que são responsáveis pela lógica de negócios, validação de dados, integração com o Firebase para autenticação e outras funcionalidades interativas.

6. **Integração com Firebase**: As últimas linhas do código importam scripts do Firebase, indicando que a aplicação utiliza Firebase para funcionalidades como autenticação de usuário. O script `firebase-init.js` é provavelmente usado para inicializar a conexão com o Firebase.

Figura 23 funcionalidade de registro de usuários

```

1  firebase.auth().onAuthStateChanged(user => {
2      if (user) {
3          window.location.href = "../home/home.html";
4      }
5  })
6
7  //Validação de email
8  function onChangeEmail() {
9      const email = form.email().value;
10     form.emailRequiredError().style.display = email ? "none" : "block";
11
12     form.emailInvalidError().style.display = validateEmail(email) ? "none" : "block";
13
14     toggleRegisterButtonDisable();
15 }
16
17 //validação de senha
18 function onChangePassword() {
19     const password = form.password().value;
20     form.passwordRequiredError().style.display = password ? "none" : "block";
21
22     form.passwordMinLengthError().style.display = password.length >= 6 ? "none" : "block";
23
24     validatePasswordsMatch();
25     toggleRegisterButtonDisable();
26 }
27
28
29 //confirmar senha
30 function onChangeConfirmPassword() {
31     validatePasswordsMatch();
32     toggleRegisterButtonDisable();
33 }
34
35
36 //registrar
37 function register() {
38     showLoading();
39
40     const email = form.email().value;
41     const password = form.password().value;
42     firebase.auth().createUserWithEmailAndPassword(
43         email, password
44     ).then(() => {
45         hideLoading();
46         window.location.href = "../../pages/home/home.html";
47     }).catch(error => {
48         hideLoading();
49         alert(getErrorMessage(error));
50     })
51 }

```

Fonte: Elaborado pelo próprio autor, 2023.

Figura 24 funcionalidade de registro de usuários

```

54 //Mensagem erro de email
55 function getErrorMessage(error) {
56     if (error.code == "auth/email-already-in-use") {
57         return "Email já está em uso";
58     }
59     return error.message;
60 }
61
62
63
64 function validatePasswordsMatch() {
65     const password = form.password().value;
66     const confirmPassword = form.confirmPassword().value;
67
68     form.confirmPasswordDoesntMatchError().style.display =
69         password == confirmPassword ? "none" : "block";
70 }
71 |
72 function toggleRegisterButtonDisable() {
73     form.registerButton().disabled = !isFormValid();
74 }
75
76 //Confirmar e cadastrar login
77 function isFormValid() {
78     const email = form.email().value;
79     if (!email || !validateEmail(email)) {
80         return false;
81     }
82
83     const password = form.password().value;
84     if (!password || password.length < 6) {
85         return false;
86     }
87
88     const confirmPassword = form.confirmPassword().value;
89     if (password != confirmPassword) {
90         return false;
91     }
92
93     return true;
94 }
95
96
97
98 const form = {
99     confirmPassword: () => document.getElementById('confirmPassword'),
100     confirmPasswordDoesntMatchError: () => document.getElementById('password-doesnt-match-error'),
101     email: () => document.getElementById('email'),
102     emailInvalidError: () => document.getElementById('email-invalid-error'),
103     emailRequiredError: () => document.getElementById('email-required-error'),
104     password: () => document.getElementById('password'),
105     passwordMinLengthError: () => document.getElementById('password-min-length-error'),
106     passwordRequiredError: () => document.getElementById('password-required-error'),
107     registerButton: () => document.getElementById('register-button')
108 }

```

Fonte: Elaborado pelo próprio autor, 2023.

Este trecho de código JavaScript, Figura 23 e Figura 24, é um componente chave da aplicação web para gerenciamento de cronograma de limpeza, especificamente para a funcionalidade de registro de usuários. Com isso temos a análise das principais funções e estruturas:

1. **Monitoramento do Estado de Autenticação**:
`Firebase.auth().onAuthStateChanged(user => {...})` é uma função do Firebase que monitora o estado de autenticação do usuário. Se um usuário estiver autenticado (`user` não é nulo), o usuário é redirecionado para a página `home.html`.

2. **Funções de Validação de Campos do Formulário**:

- `onChangeEmail()`: Valida o campo de e-mail. Verifica se o e-mail é não vazio e se atende ao formato de e-mail válido. Baseado nessas verificações, exibe mensagens de erro relevantes.

- `onChangePassword()`: Valida o campo de senha. Verifica se a senha é não vazia e se tem pelo menos 6 caracteres.

- `onChangeConfirmPassword()`: Verifica se a senha e a confirmação da senha são iguais.

3. **Registro de Usuário**:

- `register()`: Cria um novo usuário com e-mail e senha no Firebase. Se o registro for bem-sucedido, redireciona para a página `home.html`; em caso de erro, exibe uma mensagem de erro.

- `getErrorMessage(error)`: Converte códigos de erro do Firebase em mensagens legíveis para o usuário.

4. **Validação de Correspondência de Senhas**: `validatePasswordsMatch()`: Compara as senhas digitadas nos campos senha e confirmar senha, exibindo uma mensagem de erro se não coincidirem.

5. **Habilitação do Botão de Registro**:

- `toggleRegisterButtonDisable()`: Habilita ou desabilita o botão de registro com base na validade do formulário.

- `isFormValid()`: Verifica se todos os campos do formulário estão válidos (e-mail válido, senha com o tamanho mínimo e correspondência de senha).

6. **Objeto `form`**: Uma coleção de funções que acessam os elementos do DOM relacionados ao formulário, como campos de entrada e mensagens de erro.

Este código ilustra a integração da interface do usuário com a lógica de validação e autenticação. Ele utiliza Firebase para autenticação de usuário e manipulação do DOM para feedback visual.

Figura 25 estilização registro de usuário

```
1  body {
2    height: 100vh;
3  }
4  form {
5    width: 300px;
6  }
7  input {
8    border: none;
9    width: 95%;
10   padding: 10px;
11   font-size: 16px;
12   border: 1px solid #ccc;
13   border-radius: 5px;
14 }
15 [type="submit"],
16 button[class='solid'],
17 button[class='outline'] {
18   width: 100%;
19   cursor: pointer;
20   background-color: #3d9db3;
21   padding: 10px 0;
22   color: #fff;
23   font-size: 20px;
24   border: none;
25   border-radius: 5px;
26   margin-top: 20px;
27   text-shadow: 0 1px 1px #333;
28   transition: background-color 0.2s ease;
29 }
```

Fonte: Elaborado pelo próprio autor, 2023.

Este trecho de código CSS, Figura 25, é responsável por estilizar a interface de usuário da aplicação web, particularmente a página de registro de usuário. Vamos analisar cada regra de estilo e seu impacto na aparência da página:

1. **Corpo da Página (`body`)**:

- `height: 100vh;`: Define a altura do corpo da página para ser igual à altura total da janela de visualização (viewport). Isso garante que o formulário ocupe toda a altura da janela, criando uma experiência de usuário mais envolvente e focada.

2. **Formulário (`form`)**:

- `width: 300px;`: Define a largura do formulário como 300 pixels. Isso mantém o formulário centralizado e compacto, facilitando a visualização e interação em diferentes tamanhos de tela.

3. **Campos de Entrada (`input`)**:

- `border: none;`: Remove qualquer borda padrão que os navegadores possam aplicar.

- `width: 95%;`: Define a largura do campo de entrada para ocupar 95% do elemento pai (neste caso, o formulário), deixando um pouco de espaço nas laterais.

- `padding: 10px;`: Adiciona um preenchimento interno de 10 pixels para tornar a digitação mais confortável.

- `font-size: 16px;`: Define o tamanho da fonte para os textos inseridos nos campos.

- `border: 1px solid #ccc;`: Aplica uma borda sólida com cor cinza claro para definir claramente os limites do campo de entrada.

- `border-radius: 5px;`: Arredonda os cantos da borda para um design mais suave e moderno.

4. **Botões (`[type="submit"]`, `button[class='solid']`, `button[class='outline']`)**:

- `width: 100%;`: Faz com que o botão ocupe toda a largura do elemento pai.

- `cursor: pointer;`: Muda o cursor para indicar que o elemento é clicável.

- `background-color: #3d9db3;`: Define a cor de fundo do botão para um tom de azul.

- `padding: 10px 0;`: Define o preenchimento vertical do botão.

- `color: #fff;`: Define a cor do texto do botão para branco.

- `font-size: 20px;`: Aumenta o tamanho da fonte para tornar o texto do botão mais visível.

- `border: none;`: Remove a borda do botão.

- ``border-radius: 5px;``: Arredonda os cantos do botão.
- ``margin-top: 20px;``: Adiciona uma margem superior para separar visualmente o botão dos outros elementos do formulário.
- ``text-shadow: 0 1px 1px #333;``: Aplica uma sombra sutil ao texto do botão para melhorar a legibilidade.
- ``transition: background-color 0.2s ease;``: Cria uma transição suave para mudanças na cor de fundo do botão, melhorando a experiência do usuário durante a interação.

Este código CSS contribui significativamente para a estética e usabilidade da página de registro. Ele cria uma interface clara, fácil de usar e visualmente agradável, que é crucial para a experiência do usuário em aplicações web.

4.3 Avaliação 5S

A implementação do novo cronograma web de limpeza trouxe consigo uma transformação significativa na eficiência e na dinâmica da comunicação. Anteriormente, o controle do cronograma estava nas mãos dos líderes de turno e dos analistas de produção, que precisavam avaliar constantemente as necessidades de limpeza e ajustar o cronograma conforme a demanda durante suas atividades de rotina.

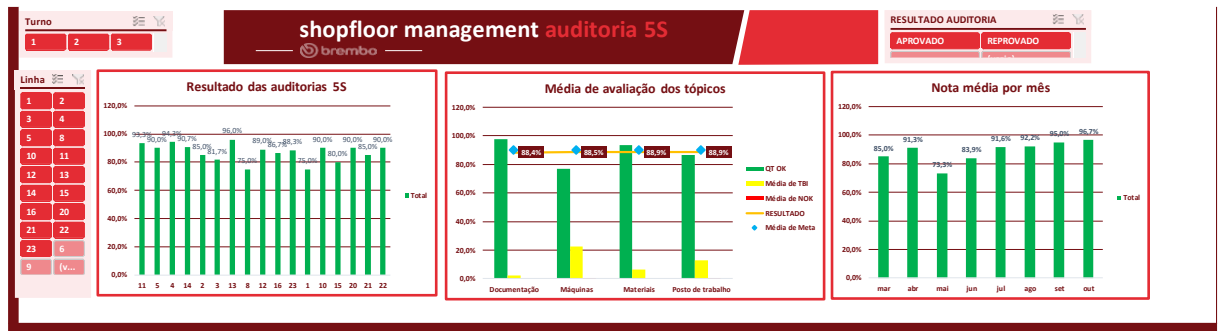
Com a introdução do cronograma web, essa tarefa tornou-se muito mais ágil e eficiente. Agora, a comunicação e a atualização do cronograma ocorrem de forma dinâmica, permitindo que os responsáveis pela programação de limpeza realizem ajustes em tempo real, garantindo que as necessidades de limpeza sejam atendidas de maneira oportuna e eficaz.

A partir da implementação do cronograma de limpeza e por meio das auditorias de 5S, observou-se uma notável melhoria nas avaliações, com uma marcante ascensão nas notas a partir de julho de 2023. Essa tendência positiva continuou nos meses subsequentes, destacando-se a agilidade na comunicação das necessidades e a adaptação dos administradores do cronograma, que incluem os líderes de turno e o analista de produção.

Essa evolução no desempenho das avaliações demonstra a eficácia do novo sistema de cronograma de limpeza, que não apenas facilitou a programação e a

execução das tarefas, mas também fortaleceu a colaboração entre os envolvidos. A abordagem dinâmica do cronograma e a rápida resposta às demandas de limpeza desempenharam um papel fundamental na melhoria contínua das práticas 5S na organização.

Figura 26 Indicador das auditorias de 5S



Fonte: Gestão de 5S *shopfloor management* Brebmo do Brasil, 2023.

Conforme ilustrado na Figura 26 Indicador das auditorias de 5S, o programa 5S na Brebmo do Brasil passou a ser monitorado por meio de auditorias de gestão da produção a partir de março de 2023, quando a pontuação inicial era de 91,3%. Durante esse período, foram implementadas ações de melhoria, incluindo a contratação de mais mão de obra para limpeza em abril, o que resultou na manutenção da pontuação em 91,3%. No entanto, com o aumento da produção, a comunicação tornou-se mais lenta, uma vez que outras atividades tiveram prioridade, resultando em uma leve queda.

A situação começou a se recuperar em junho, principalmente devido às ações tomadas, incluindo visitas de clientes. No entanto, foi a partir de julho, com a implementação do novo cronograma, que a capacidade de apresentar as reais necessidades de limpeza no sistema se tornou uma realidade. Mesmo diante de uma alta demanda de produção, o novo sistema permitiu uma comunicação mais eficaz e o acesso simultâneo a informações essenciais, sem perda de tempo.

5 CONCLUSÃO

Neste trabalho, foram abordadas diversas características de um documento de especificação de software. A partir da análise realizada, identificamos alguns problemas e, como solução, desenvolvemos um sistema de gerenciamento que visa aprimorar a comunicação entre os envolvidos, incluindo líderes de turno, analistas e auxiliares de limpeza.

O foco principal deste projeto foi o desenvolvimento de uma aplicação web, na qual aplicamos técnicas e linguagens de programação para a internet. No sistema resultante, todos os recursos são acessados por meio de aplicativos que permitem a navegação na internet.

Foi utilizado tecnologias como HTML, JavaScript e CSS para criar um produto que não apenas oferece uma interface visual agradável e intuitiva, mas também é responsivo, adaptando-se ao tamanho da tela do dispositivo utilizado.

Após a implementação do sistema em um ambiente real e sua utilização, o acompanhamento do cronograma de limpeza tornou-se mais ágil e flexível. Os usuários podem acessar as informações a partir de diversos dispositivos, desde que tenham conexão à internet. Relatos dos usuários destacaram o conforto proporcionado pelo website, especialmente a agilidade na resposta da equipe de limpeza em relação às necessidades sinalizadas pela liderança. Ao final de uma semana de trabalho, há um registro completo das modificações realizadas ao longo do tempo. Além disso, o novo sistema eliminou a necessidade de procurar pessoalmente os responsáveis pela execução das atividades ou o envio de e-mails para coordenar tarefas futuras.

A Tabela 1 apresenta algumas comparações relevantes após a implementação do novo sistema desenvolvido.

Tabela 1 comparativo de sistemas

	Sistema desenvolvido	sistema utilizado anteriormente
Acesso por dispositivos móveis	Sim	Não
Acesso do cronograma pelos auxiliares de limpeza	Sim	Apenas arquivo impresso
Modificação do cronograma	analista e líderes	Não
Necessidade do e-mail corporativo para acesso	não	Sim
Necessidade de download	não	Sim
Arquivos físicos	não	Sim

Fonte: Elaborado pelo próprio autor, 2023.

O sistema desenvolvido oferece a flexibilidade de gerenciar o cronograma a partir de qualquer dispositivo com acesso à internet. Atualmente, o sistema é hospedado no domínio do GitHub, e há um estudo detalhado para que seja conduzido sobre a sua distribuição em todas as unidades do Grupo Brembo.

5.1 Problemas Encontrados

Durante o processo de desenvolvimento desta aplicação web para gerenciamento de cronograma de limpeza, foi enfrentado vários desafios técnicos e conceituais. Um dos obstáculos mais significativos foi relacionado ao armazenamento de dados, especificamente a integração e gestão de dois diferentes sistemas de banco de dados. Essa complexidade surgiu devido à necessidade de armazenar e manipular conjuntos de dados distintos, que eram essenciais para o funcionamento eficaz da aplicação.

A integração de múltiplos bancos de dados apresentou um desafio notável, especialmente no que diz respeito à consistência, integridade e eficiência dos dados. Foi uma tarefa árdua garantir que ambos os sistemas trabalhassem de forma harmoniosa, mantendo o desempenho e a confiabilidade. Além disso, essa dificuldade interferiu na implementação completa de uma funcionalidade crucial: uma página dedicada à gestão de auditoria de 5S dentro da própria aplicação. O 5S, sendo um método amplamente reconhecido para otimização de processos e organização do

ambiente de trabalho, era um componente essencial para a aplicação, visando aprimorar a gestão de limpeza.

A ausência dessa funcionalidade integral na versão atual da aplicação destacou a necessidade de desenvolver um método eficiente para incorporar a gestão de auditoria de 5S. Este aspecto tornou-se um foco primordial para futuras melhorias. A integração dessa funcionalidade não só ampliaria a abrangência da aplicação, mas também ofereceria uma ferramenta valiosa para os usuários, facilitando o monitoramento e a melhoria contínua dos processos de limpeza e organização.

Portanto, a solução para este desafio envolve um trabalho mais aprofundado na arquitetura de dados e na integração de sistemas, possivelmente explorando novas tecnologias ou métodos de banco de dados para superar as limitações atuais. Além disso, a implementação de uma interface de usuário intuitiva e eficiente para a gestão de auditoria de 5S será crucial para garantir que a funcionalidade seja não apenas tecnicamente viável, mas também prática e benéfica para os usuários finais.

A superação desses obstáculos será um passo significativo para o aprimoramento da aplicação, contribuindo para um gerenciamento de limpeza mais eficiente e estruturado, alinhado com os princípios do 5S. Como resultado, a aplicação não apenas atenderá melhor às necessidades de seus usuários, mas também se destacará como uma ferramenta inovadora no campo de sistemas de gerenciamento de limpeza.

5.2 Trabalhos futuros

É importante refletir sobre as oportunidades de crescimento e melhorias que podem ser implementadas na aplicação web de gerenciamento de cronograma de limpeza. Um aspecto crucial que ainda não foi incorporado na aplicação atual é a funcionalidade de auditoria de 5S. Atualmente, a auditoria de 5S ainda é realizada utilizando planilhas do Excel, o que representa uma área significativa para futuras melhorias na aplicação.

A auditoria de 5S, um método reconhecido para otimizar a eficiência e organização em ambientes de trabalho, é uma ferramenta essencial para a gestão e avaliação da limpeza e da ordem. A integração dessa funcionalidade na aplicação web propiciaria uma maneira mais eficaz e automatizada de realizar estas auditorias,

substituindo o processo manual e mais suscetível a erros que é realizado atualmente no Excel.

Trabalhos futuros devem se concentrar no desenvolvimento e na implementação de um módulo de auditoria de 5S dentro da aplicação. Isso incluiria a criação de uma interface de usuário intuitiva que permita aos usuários registrarem, acompanhar e analisar os resultados da auditoria diretamente na aplicação. Além disso, seria importante garantir a integração desse módulo com os demais componentes da aplicação, permitindo uma visão holística e integrada do processo de limpeza e manutenção.

A implementação dessa funcionalidade também envolverá superar desafios técnicos, tais como a adaptação da arquitetura de dados para suportar as informações adicionais geradas pelas auditorias de 5S e garantir que a aplicação possa processar e apresentar esses dados de maneira eficiente. Isso poderia incluir o desenvolvimento de novas funcionalidades, como relatórios personalizáveis, dashboards interativos e alertas automáticos baseados nos resultados das auditorias.

Finalmente, a transição da auditoria de 5S do Excel para a aplicação web traria uma série de benefícios, como a redução de erros manuais, a melhoria na acessibilidade e compartilhamento de informações, e a possibilidade de análises mais profundas e insights baseados em dados. Este avanço representaria um passo significativo na direção de tornar a aplicação uma ferramenta mais abrangente e eficiente para a gestão de cronogramas de limpeza, alinhando-se com as práticas modernas de gestão de qualidade e eficiência operacional.

5.3 Inclusão de Links na Aplicação

A integração de links diretos para as instruções de trabalho na aplicação web representa uma inovação significativa no acompanhamento das tarefas de limpeza. Essa funcionalidade permite que os usuários acessem rapidamente as informações necessárias para executar corretamente as atividades de limpeza, sem a necessidade de buscar manuais impressos ou consultar supervisores. Ao clicar em um link associado a uma tarefa específica, o usuário é direcionado a uma página detalhada com as instruções de trabalho, que podem incluir textos explicativos, imagens ilustrativas e até vídeos instrutivos.

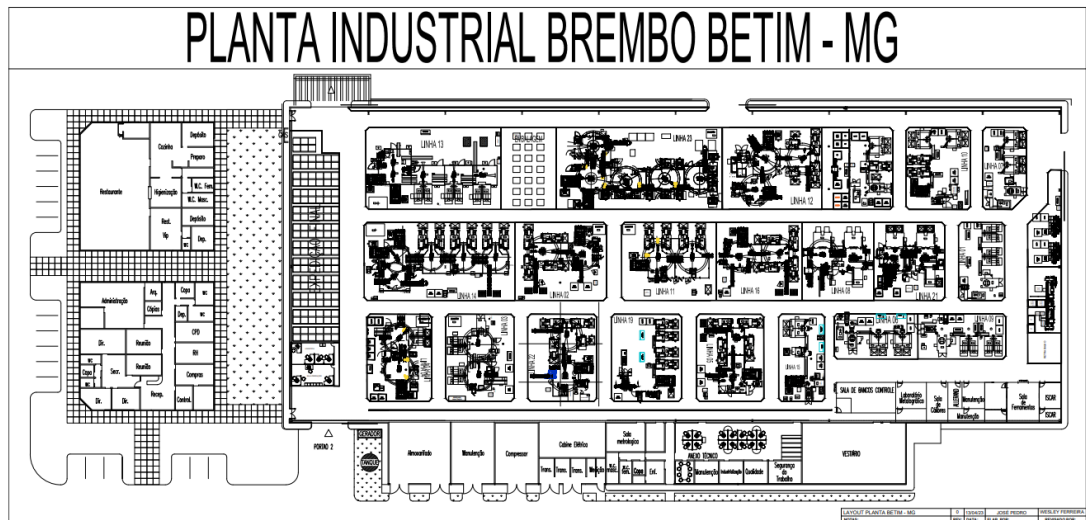
A disponibilidade desses recursos diretamente na aplicação não apenas facilita a execução das tarefas de limpeza, mas também promove a padronização das práticas de limpeza em toda a indústria. Isso garante que todos os colaboradores sigam os mesmos procedimentos, resultando em uma limpeza mais eficiente e consistente. Além disso, a possibilidade de incluir links para outros recursos relevantes, como fichas de segurança de produtos químicos ou manuais de equipamentos, amplia ainda mais a utilidade da aplicação como uma ferramenta abrangente para o gerenciamento de limpeza.

5.4 Inclusão de um Modelo de Planta Baixa

A incorporação de um modelo de planta baixa na aplicação web é uma estratégia eficaz para visualizar a distribuição das áreas de trabalho e a localização das tarefas de limpeza dentro da fábrica. Essa representação gráfica permite que os usuários identifiquem rapidamente as áreas específicas onde as atividades de limpeza devem ser realizadas, facilitando o planejamento e a organização do trabalho. Ao selecionar uma área na planta baixa, o usuário pode visualizar uma lista de tarefas de limpeza associadas a essa área, bem como as instruções de trabalho correspondentes.

A utilização de uma planta baixa setorizada também é fundamental para a gestão eficiente dos recursos de limpeza. Com essa ferramenta, os supervisores podem alocar equipes e materiais de limpeza de forma mais direcionada, evitando desperdícios e otimizando o tempo de trabalho. Além disso, a planta baixa pode ser atualizada para refletir mudanças na disposição das áreas de trabalho ou na distribuição das tarefas de limpeza, garantindo que a aplicação permaneça como um recurso atualizado e confiável para o gerenciamento da limpeza na indústria.

Figura 27 – Planta Brembo betim



Fonte: Brembo do Brasil, 2024.

5.5 Criação de Indicadores Futuros

A implementação da aplicação web para o acompanhamento das tarefas de limpeza na indústria proporciona uma base de dados rica e detalhada que pode ser utilizada para a criação de indicadores de desempenho futuros. Esses indicadores são ferramentas valiosas para avaliar a eficiência das práticas de limpeza e identificar áreas que necessitam de melhorias. Por exemplo, a partir das informações coletadas, é possível desenvolver indicadores como o tempo médio de execução de tarefas, a frequência de limpezas realizadas em determinadas áreas e o cumprimento dos cronogramas estabelecidos.

Além disso, a análise dos dados permite a identificação de tendências e padrões no processo de limpeza. Por exemplo, se determinada área da fábrica apresenta frequentemente atrasos na execução das tarefas de limpeza, isso pode indicar a necessidade de ajustes na alocação de recursos ou na programação do trabalho. Da mesma forma, a comparação dos indicadores de desempenho ao longo do tempo pode revelar o impacto de mudanças nas práticas de limpeza, como a introdução de novas técnicas ou equipamentos.

A utilização desses indicadores de desempenho não se limita apenas à avaliação da limpeza, mas também contribui para a melhoria contínua dos processos industriais. Eles fornecem insights valiosos para a tomada de decisões estratégicas, como a otimização de recursos, a implementação de treinamentos para os

colaboradores e a definição de metas de limpeza mais desafiadoras. Além disso, a comunicação transparente dos indicadores de desempenho aos colaboradores pode incentivar o engajamento e a responsabilização de todos os envolvidos na manutenção da limpeza e da organização da indústria.

Portanto, a criação de indicadores futuros a partir das informações geradas pela aplicação web é um passo fundamental para a gestão eficaz da limpeza na indústria. Esses indicadores permitem não apenas a avaliação do desempenho atual, mas também o planejamento de ações para aprimorar as práticas de limpeza e promover um ambiente de trabalho mais seguro, produtivo e sustentável.

REFERÊNCIAS

Flanagan, D. (2011). JavaScript: The Definitive Guide. O'Reilly Media, Inc.

Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.

Thiago Leão. “5S - O Que é 5S E Como Implementar O Programa Na Sua Empresa.” Blog Industrial Nomus, 17 Mar. 2020, www.nomus.com.br/blog-industrial/5s/. Acesso em: 5 abril de 2023.

ARAÚJO, L. C. de. 5S: Housekeeping e qualidade total. 2. ed. rev. e atual. Rio de Janeiro: Qualitymark Editora, 2007.

FARIA, L. G. de. Desenvolvimento web com PHP e MySQL. 5. ed. Rio de Janeiro: Elsevier, 2017.

Google Chrome. Google. Disponível em: <https://www.google.com/chrome/>. Acesso em: 14 dez. 2023.

Firestore. Google. Disponível em: <https://Firestore.google.com/>. Acesso em: 14 dez. 2023.

GITHUB. Consulta. Disponível em: <https://github.com/search?o=desc&q=stars%3A%3E1&s=stars&type=Repositories>> Acesso em: 12 de junho de 2023.

SILVA, A. B. da. Interface com o usuário: conceitos e técnicas. 2. ed. São Paulo: Novatec Editora, 2015.

ROCHA, L. M. da. Segurança de informação em sistemas web. 3. ed. Rio de Janeiro: Elsevier, 2018.